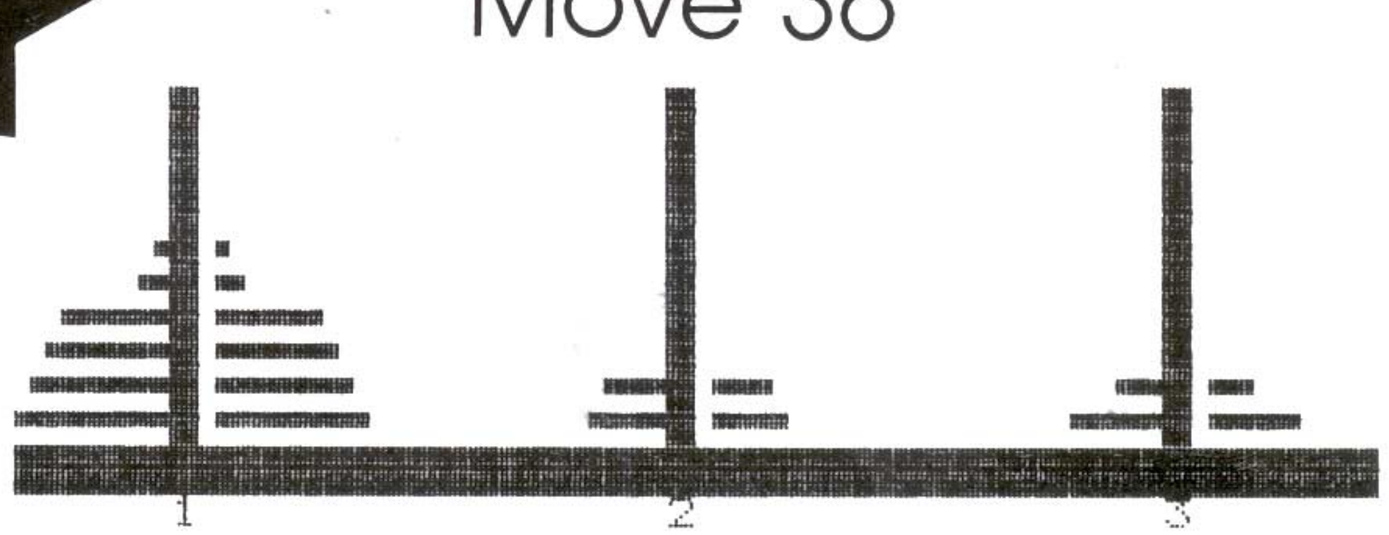
TRS-80 SYSTEM 80 VIDEO GENIE PMC-80

Issue 24, November 1981

NEW FREE SOFTWARE OFFER INSIDE!



THE TOWERS OF HANOI

Also in this issue:

HARDWARE:

Add Joysticks and Input/Output Ports to your '80 — Part 4

PROGRAMMING:

The Theory and Techniques of Sorting — Part 2

SOFTWARE:

- Level 1 Tic Tac Toe
- Multiple Regression Analyser
- BASIC Line Reference Validator
- Learn Your Atomic Tables
- Attack
- Text Typer

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034. AUSTRALIA. TELEPHONE (08) 211 7244. PRICE: AUS. $2.50, N.Z. $4.00, U.K. £1.50

***** ABOUT MICRO-80 *****

EDITOR:	IAN VAGG
SOFTWARE EDITOR:	CHARLIE BARTLETT
HARDWARE EDITOR:	EDWIN PAAY
U.K. CORRESPONDENT:	TONY EDWARDS

MICRO-80 is an international magazine devoted entirely to the Tandy TRS-80 microcomputer and the Dick Smith System 80/Video Genie. It is available at the following prices (all prices shown in Aus.$ except for U.K. prices which are in pounds Sterling).

12 months subscription	Aus.	$24.00
	NZ.	$36.00 (Airmail)
	Hong Kong	$46.00 (Airmail)
	U.K.	£16.00
Single Copy	Aus.	$2.50
	N.Z.	$3.50 (Airmail)
	Hong Kong	$4.25 (Airmail)
	U.K.	£1.50
Months programs on cassette	Aus.	$3.50
	N.Z.	$4.00 (Airmail)
	Hong Kong	$4.50 (Airmail)
(at present available from Australia only)	U.K.	$4.75 (Airmail)
12 months subscription to magazine and cassette	Aus.	$60.00
	N.Z.	$78.00 (Airmail)
	Hong Kong	$88.00 (Airmail)
	U.K.	£41.00 (Airmail)

Special bulk purchase rates are also available to computer shops etc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80, System 80 or Video Genie and their peripherals. MICRO-80 is in no way connected with either the Tandy or Dick Smith organisations.

** WE WILL PAY YOU TO PUBLISH YOUR PROGRAMS **
Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your TRS-80 or System 80 to earn some extra income is included in every issue.

** CONTENT **
Each month we publish at least one applications program in Level I BASIC, one in Level II BASIC and one in DISK BASIC (or disk compatible Level II). We also publish Utility programs in Level II BASIC and Machine Language. At least every second issue has an article on hardware modifications or a constructional article for a useful peripheral. In addition, we run articles on programming techniques both in Assembly Language and BASIC and we print letters to the Editor and new product reviews.

** LIABILITY **
The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

***** CONTENTS *****

MICRO-80 is registered by Australia Post - Publication SQB 2207 Category B

AUSTRALIAN OFFICE AND EDITOR:

MICRO-80, P.O. BOX 213, GOODWOOD, SOUTH AUSTRALIA, 5034. TEL. (08) 211 7244

U.K. SUBSCRIPTION DEPT:

24 WOODHILL PARK, PEMBURY, TUNBRIDGE WELLS, KENT TN2 4NW.

Printed by:

Shovel & Bull Printers, 312A Unley Road, HYDE PARK, S.A. 5061

Published in Australia by MICRO-80, 433 Morphett Street, ADELAIDE.

***** EDITORIAL *****

Despite the promises, very few of the new colour computers have actually been seen in the shops yet. Commodore with its VIC-20 and Tandy with the TRS-80 Colour Computer, are both resigned to missing the Christmas rush but it does seem that these computers will actually be available in February, 1982.

So, with Tandy at long last about to have a colour computer, what of the Dick Smith organisation? Whilst there have been rumours of colour add-ons for the System 80/Video Genie, there have been no suggestions that the manufacturers of the Video Genie actually have a colour computer on the way. So, Dick Smith has solved his problem himself and will shortly be competing with Tandy in offering colour computers. How has he done it? DICK SMITH ELECTRONICS WILL SHORTLY BE SELLING THE VIC-20 BY COMMODORE. It will be a fascinating battle to see which computer gains the most customers. The VIC-20 in its most basic form wins over the TRS-80 Colour on price - $399 versus $599, but before a judgement can be made, it is necessary to compare the prices for equipment of similar performance (both systems have a wide range of add-ons and expansion units). That we will do for you in the next few issues. Next month we will have a detailed review of the TRS-80 Colour Computer, based on hands-on experience of a reader who imported his own several months ago, and our own impressions of an expanded BASIC machine loaned to us by Tandy. We will follow that in a later issue with a comparative review of the VIC-20. We will also look at one of the newest offerings, the Hitachi Peach. There are exciting times ahead for those contemplating the purchase/replacement of a computer, and we will do our part in keeping you informed.

This issue marks the completion of two years of publication for MICRO-80. The next 12 months promise to be even more exciting in microcomputing than the past. We intend to rise to the challenge in a number of ways. We will expand our coverage of the familiar computers and, in particular, will give increased support to the Model III. Not only will the magazine carry a lot of hitherto unpublished information about the internals of the Model III, but Eddy Paay is putting the finishing touches to his Model III ROM Reference Manual and a great Debug utility program for the Model III. Many more of our readers now operate disk systems so we will include much more information and programs for disk users. Then there are the colour computers. It is our intention to support the TRS-80 Colour Computer, at least. This will not be done at the expense of the space given to the TRS-80 Models I and III and the System 80, rather we will increase the size of the magazine to accommodate the extra material.

NEW PROGRAM LIBRARY AVAILABLE FREE TO MICRO-80 SUBSCRIBERS.

We have developed a new program library of excellent quality and worth over $100 at retail value. The seven new programs on cassette or disk, together with a comprehensive and informative manual, will be sent free of charge to all new subscribers and to those who renew their subscription from Issue 25 (December, 1981) onwards. Further details of the programs contained in our new software library will be published shortly.

INFLATION STRIKES (BUT JUST A LITTLE).

The price for a 12 months' subscription to MICRO-80 has remained unchanged since we commenced publication 2 years ago. During that time, the Consumer Price Index in Australia has increased by more than 20%. Commencing next issue, the price of a subscription to MICRO-80 magazine will increase to $26, a Cassette subscription to $65 and a Disk subscription to $125. These increases are very much less than the inflation rate and the increased revenue will assist us to improve the quality of your magazine still further.

DIGITAL COMMUNICATIONS COMING TO AUSTRALIAN MICRO-USERS.

One of the more exciting uses for a microcomputer is to use it as a communications medium, i.e. for one microcomputer to communicate with another several kilometers distant. Many of our readers have already tried sending programs over the telephone line by connecting amplifiers to the cassette port, we know. This is an unsophisticated version of an acoustic coupler which is widely used for transmitting digital information between commercial computers.

One method of implementing automatic data processing in a business is to use a time-sharing bureau via a dial-up line. Turning that jargon into some semblance of English, we find that an alternative to installing your own computer is to buy a terminal and printer and connect to a large computer via an ordinary telephone line. The large computer runs your programs in response to commands from the terminal and you can produce listings, reports, etc. on your printer. The large computer is shared by many other users which keeps the cost to each user down to an acceptable level. These computers are owned by companies called bureaus. The volume of data being transferred around the world at any one time is large and growing and is significant with respect to the amount of voice traffic handled by telephone lines. Many bureaus have computers in only one or two locations in the country and users' data is transferred by long distance telephone lines (usually high speed lines). In the extreme, the computer you are using might actually be on the other side of the world and your data is beamed around the globe via a satellite.

All of the above is happening now and is expensive. One of the problems that bureaus have is

that most businesses want to process data during normal working hours, so large (expensive) computers and peripherals need to be installed to cope with peak traffic whilst, for at least half the time, these machines are very much under-utilised. Even attractive, low, off-peak rates fail to keep them fully utilised since it is usually prohibitively expensive for their customers to bring in staff at night to process data.

About 18 months ago, an enterprising bureau in the U.S.A. had a bright idea. What if they reduced their off-peak rates way, way down, put some useful BASIC programs on their computers and sold off-peak time to personal computer users, most of whom were working for their employees in peak times anyway? Thus was born Micronet. Others were quick to follow. Electronic mail systems and bulletin boards abound. These are systems whereby computer owners across the country communicate with each other via the telephone lines and leave messages on disks driven from the host computer.

Some more enterprising souls amongst the Australian microcomputing fraternity have joined their American colleagues by using the MIDAS link provided by the Overseas Telecommunication Company (OTC). Unfortunately, this can be very expensive indeed, possibly costing as much as $50-$60 per hour, depending upon the amount of data transferred through the link. Help is at hand, however. An Australian microcomputer network has now been announced, called the AUSTRALIAN SOURCE, (no doubt after one of the most successful similar U.S. organisations, the SOURCE but, we understand, there is no direct relationship between the two). The AUSTRALIAN SOURCE will commence operations in Melbourne early in 1982. Thereafter, it will become available in Sydney about 30 days later then Brisbane, Perth and Adelaide in that order so that, by about the middle of 1982, users in all major capital cities will be able to link up with the AUSTRALIAN SOURCE via a local telephone call.

There is an initial fee of $100 to become an AUSTRALIAN SOURCE subscriber but founder subscriptions taken out prior to 1st February, 1982 can be had for only $60. Thereafter, users pay $10 per hour in peak time (8 a.m. - 6 p.m.) and $4.50 per hour in off-peak time (6 p.m. - 8 a.m.). MICRO-80 is very excited about the possibilities which this new venture opens to microcomputer owners. We will support the AUSTRALIAN SOURCE will articles explaining how to adapt your equipment to communicate with the SOURCE and membership of the SOURCE is available from our mail order department. In fact, we have reserved a number of charter memberships for our readers and we have also taken the unusual step (for us) of accepting an advertisement in our columns for the AUSTRALIAN SOURCE which gives you further information about this service (see back cover).

As we said above, we will discuss the equipment requirements in more detail in a future issue. The minimum requirement, however, will be a duplex serial port able to operate at 300 baud and an acoustic coupler. At present an '80 user would require an expansion interface to meet the requirement for a serial port. We are looking at a cheaper alternative for those without expansion interfaces - more news in later issues.

- 0000000000 -

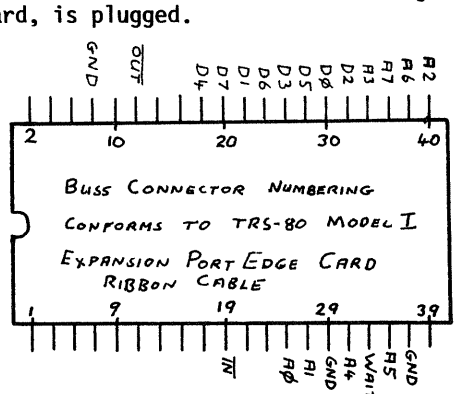
***** JOYSTICKS AND INPUT/OUTPUT PORTS FOR YOUR '80 by Alan Dent *****

PART 4 - SOFTWARE DRIVERS

In this concluding part we will look at several driver programs to enable you to use both the joysticks and input/output ports. We had also intended to include full-sized masters for the PC board in this part. However, one or two tracks need to be added/changed and we have held over publication until the revised artwork is available (hopefully next issue). We will also announce the price of the PC board itself, in that issue.

Last month we included two circuit diagrams on Page 12, without comment. These were for a port output monitor and a port input switch. The output monitor enables you to visually detect the status of each bit of an output port. Any bit having a value of 1 will cause its LED to be "ON". A LED will be off when its corresponding bit has a value of 0. The port input switch enables you to set up any desired bit pattern on an input port. You may like to check the operation of your ports by writing a simple program to transfer the byte on the port input switch to an output port. Each bit that is "ON" at the input should also be "ON" at the output.

Finally, the diagram below shows the buss connector numbering for the socket into which the ribbon cable to the TRS-80 keyboard, is plugged.



The first listing below contains three simple demonstration programs which enable you to move a dot around the screen, draw lines etc., using the joysticks. The first joystick program simply inputs the digital value of the voltage at the joystick moving arms. This is then processed to conform to the constraints of the level 2 X,Y maximums of 127,47 respectively. The graphic point location is then SET and RESET to flash the position of the joystick controlled graphic point. The 8 bit values of the two inputs are also displayed in decimal at the bottom left corner of the screen. The next program, accessed by RUN 1ØØ, is similar but does not RESET and hence draws a continuous line as the joystick is moved. The third joystick program, RUN 2ØØ, combines the actions of both of the preceding programs. You can draw lines with the addition that if the space bar is pressed, the spot can be moved while flashing but not drawing, to a different location, ready to start drawing again. This one is a good one for young kids to play with.

```
5 REM  *** THIS PROGRAM MOVES A SPOT AROUND THE SCREEN
        LINE 10 CLEARS THE SCREEN
        LINE 20 INTEGERS ARE FASTER
        LINE 30 FETCHES JOYSTICKS DIGITAL VALUE F
6 REM   LINE 40 TURNS OFF THE GRAPHIC POINT
        LINE 50 LIMITS "X" TO 127
        LINE 60 LIMITS "Y" TO 47
        LINE 70 TURNS ON GRAPHIC POINT
        LINE 80 PRINTS JOYSTICK DIGITAL VALUE
        LINE 90 LOOPS BACK TO FETCH NEW VALUES
10 CLS
20 DEFINT A-Z
30 A=INP(0) : B=INP(1)
40 RESET (X,Y)
50 X=A/2
60 Y=B/5 : IF Y>47 THEN Y=47
70 SET(X,Y)
80 PRINT@960,"X=";A;"  ","Y=";B;"  ";
90 GOTO30
95 REM ************************************************************
100 REM THIS PROGRAMME DRAWS LINES ON SCREEN WITH JOYSTICKS
105 CLS
110 DEFINT A-Z
120 A=INP(0)
130 B=INP(1)
140 X=A/2
150 Y=B/5 : IF Y>47 THEN Y=47
160 PRINT@960,"X=";A;"  ","Y=";B;"  ";
170 SET (X,Y)
180 GOTO120
190 REM ************************************************************
200 REM THIS PROGRAMME DRAWS LINES ON THE SCREEN OR MOVES
        A SPOT AROUND THE SCREEN IF THE SPACE BAR IS PRESSED
205 CLS
210 X=INP(0)/2 : Y=INP(1)/5 : IF Y>47 THEN Y=47
220 SET(X,Y)
230 IF PEEK(14400)=128 THEN RESET(X,Y) ' TEST FOR SPACE BAR
240 GOTO 210
290 REM ************************************************************
300 REM THIS PROGRAMME SETS THE "PPI" TO PORT 8 IN, 9 & 10 OUT
        READS THE SWITCHES AT THE INPUT TO PORT 8, PRINTS
        THE DECIMAL AND HEX VALUE TO THE SCREEN AND OUTPUTS
        THE VALUE TO PORT 9
305 CLS : PRINTCHR$(23) : OUT(11),152
310 X=INP(8)
320 PRINT@530,X;"   "
325 REM LINES 330/340 CONVERT THE DECIMAL VALUE TO HEX
330 A=INT(X/16): IF A<10 THEN A$=CHR$(A+48) ELSE A$=CHR$(A+55)
340 B=X-A*16 : IF B<10 THEN B$=CHR$(B+48) ELSE B$=CHR$(B+55)
350 C$=A$+B$
360 PRINT@550,C$;"  "
370 OUT (9),X
380 GOTO 310
390 REM ************************************************************
400 ' THIS PROGRAMME SCANS THE KEYBOARD AND OUTPUTS THE ASCII
        VALUE TO THE LED DISPLAY AT PORT 9.
        IT ALSO READS THE SWITCHES AT THE INPUT TO PORT 8 AND
        DISPLAYS THE COMPUTER CODE IT REPRESENTS ON THE SCREEN
405 CLS : PRINTCHR$(23) : OUT(11),152
410 A$=INKEY$ : IFA$="" THEN 440
420 X=ASC(A$)
430 OUT(9),X
```

```
440 X=INP(8)
450 IF X=Y THEN 410 ELSE Y=X
460 IF X>191 THEN X$="SPACE COMPRESSION CODE" ELSE IF X>127 THEN X$="GRAPHIC COD
E  "+CHR$(X) ELSE IF X>31 THEN X$="CHARACTER  "+CHR$(X) ELSE X$="CONTROL CODE"
470 CLS:PRINTCHR$(23)
480 PRINT@0,X$
490 GOTO 410
```

We now move on to the parallel I/O ports. At RUN 300 we first program the PPI at address location 8-11 by writing the control word decimal 152 to port 11, the control register location. Port 8 is then read to input the value of the switches connected to it. The decimal value and the HEX value is displayed on the screen and then written to port 9 to switch on the LED display being controlled by that port. The next program starting at 400, scans the keyboard and outputs the ASCII value to the LED's at port 9. It then reads the switches at the input to port 8 and prints on the screen the computer code that the switches represent. Rod Stevenson has written a machine language program which also allows you to draw pictures on the screen under joystick control. It is more elegant than mine as my speciality is in hardware and Rod has demonstrated his software ability with his series "BETTER BASIC PROGRAMMING". Rod's program is also included in the program listings in two versions. One is an EDTASM source listing, the other is poked in from a BASIC program.

```
                00100 ;
                00110 ;WRITTEN 29/9/81 BY ROD STEVENSON TO USE JOYSTICK KIT
                00120 ;DEVELOPED BY ALLAN DENT FOR ADELAIDE USERS' GROUP.
                00130 ;
                00140 ;L-KEY WILL DRAW LINE (DISABLES RESET)
                00150 ;D-KEY WILL MAKE DOT (TO ENABLE CURSOR TO BE MOVED)
                00160 ;SHIFT UP ARROW RETURNS TO BASIC
                00170 ;CLEAR-KEY WILL CLEAR SCREEN.
                00180 ;
                00190 ;MAY BE RELOCATED BY CHANGING ORG STATEMENT
                00200 ;THERE IS A VERSION CONVERTED TO BASIC POKE STATEMENTS.
                00210 ;
7F00            00220           ORG     32512
7F00 CD7F0A     00230 USRVAL    CALL    0A7FH   ;GET VALUE FROM USR(0)
7F03 7D         00240           LD      A,L
7F04 32647F     00250           LD      (VALUE),A;SAVE VALUE FOR LATER
7F07 CD5B03     00260 INKEY     CALL    35BH    ;INKEY ROUTINE
7F0A FE1B       00270 RETN      CP      01BH    ;TEST IF SHIFT UP ARROW
7F0C 2006       00280           JR      NZ,CLS  ;IGNORE IF NOT
7F0E 2A657F     00290           LD      HL,(COORD);COORDS TO BASIC
7F11 C39A0A     00300           JP      0A9AH   ;BACK TO BASIC
7F14 FE1F       00310 CLS       CP      01FH    ;TEST IF CLEAR PRESSED
7F16 CCC901     00320           CALL    Z,1C9H  ;CLS ROUTINE
7F19 21647F     00330 LINE      LD      HL,VALUE;CHECK IF L PRESSED
7F1C FE4C       00340           CP      'L'
7F1E 2002       00350           JR      NZ,DOT  ;IGNORE IF NOT
7F20 365A       00360           LD      (HL),90
7F22 FE44       00370 DOT       CP      'D'     ;CHECK IF D PRESSED
7F24 2002       00380           JR      NZ,YCOORD;IGNORE IF NOT
7F26 3664       00390           LD      (HL),100
7F28 DB01       00400 YCOORD    IN      A,(1)   ;GET Y VALUE
7F2A 06FF       00410           LD      B,0FFH  ;USE B AS COUNTER FOR DIVIDE
7F2C 04         00420 DIVD5     INC     B       ;DIVIDE Y COORD BY 5
7F2D D605       00430           SUB     5
7F2F 30FB       00440           JR      NC,DIVD5;SUBTRACT TILL <=0
7F31 78         00450 GR47      LD      A,B     ;ANSWER INTO A TO
7F32 FE2F       00460           CP      47      ;CHECK IF >47
7F34 3802       00470           JR      C,NGR47 ;IGNORE IF NOT
7F36 3E2F       00480           LD      A,47    ;IF > 47 LET = 47
7F38 6F         00490 NGR47     LD      L,A     ;L HAS Y COORD.
7F39 DB00       00500 XCOORD    IN      A,(0)   ;GET X VALUE
7F3B CB3F       00510           SRL     A       ;DIVIDE BY 2
7F3D 67         00520           LD      H,A     ;H HAS X COORD.
7F3E 22657F     00530           LD      (COORD),HL;SAVE COORDS FOR BASIC
7F41 44         00540 ROM       LD      B,H     ;PREPARE FOR ROM ROUTINE
7F42 7D         00550           LD      A,L
7F43 E5         00560           PUSH    HL      ;SAVE LOCATION FOR RESET
7F44 2680       00570           LD      H,80H   ;SET FLAG FOR ROUTINE
7F46 CD4B7F     00580           CALL    GRAFIX  ;ROM ROUTINE
7F49 1808       00590           JR      RESET
7F4B E5         00600 GRAFIX    PUSH    HL
7F4C C5         00610           PUSH    BC
7F4D 218C18     00620           LD      HL,188CH
```

```
7F50 C35001    00630          JP      150H
7F53 E1        00640 RESET    POP     HL        ;GET LOCATION
7F54 3A647F    00650          LD      A,(VALUE);VALUE OF USR(0)
7F57 FE64      00660          CP      100       ;CHECK IF RESET REQUIRED
7F59 20AC      00670          JR      NZ,INKEY;RESET IF FLAG 100
7F5B 44        00680          LD      B,H       ;PREPARE FOR ROM ROUTINE
7F5C 7D        00690          LD      A,L
7F5D 2601      00700          LD      H,01      ;RESET FLAG FOR ROUTINE
7F5F CD4B7F    00710          CALL    GRAFIX    ;ROM ROUTINE
7F62 18A3      00720          JR      INKEY
7F64 00        00730 VALUE    DEFB    00        ;SPACE FOR USR(0) VALUE
7F65 0000      00740 COORD    DEFW    00        ;SPACE FOR COORDS.
7F00           00750          END     USRVAL
```

```
20 POKE16526,32512AND255:POKE16527,INT(32512/256):CLEAR50:POKE16
561,32512AND255:POKE16562,INT(32512/256)      'FIRST SET PASSES
LOCATION TO USR(0), SECOND SET PROTECTS MEMORY. CHANGE THESE IF
YOU RE-ASSEMBLE ROUTINE IN A DIFFERENT MEMORY LOCATION.
30 CLS:PRINT@448,"THIS ROUTINE ACCEPTS INPUT FROM THE JOYSTICK D
EVELOPED BY THE   ADELAIDE USERS' GROUP. IT REMAINS IN M/L UNTIL
 SHIFT-UP ARROW ISPRESSED. CLEAR-KEY WILL CLS. L-KEY WILL DRAW A
 LINE (DISABLE    RESET). D-KEY WILL DRAW A DOT (ENABLE RESET)."
40 FORI=32512TO32615:READD:POKEI,D:NEXT
50 DATA205,127,10,125,50,100,127,205,91,3,254,27,32,6,42,101,127
,195,154,10,254,31,204,201,1,33,100,127,254,76,32,2,54,90,254,68
,32,2,54,100,219,1,6,255,4,214,5,48,251,120,254,47,56,2,62,47,11
1,219,0,203,63,103,34,101,127,68,125
60 DATA229,38,128,205,75,127,24,8,229,197,33,140,24,195,80,1,225
,58,100,127,254,100,32:  'TO RESET UNLESS 100 IS PASSED, CHANGE
THIS LAST DATA ITEM (NOW 32) TO 40. HOWEVER, THIS WILL ALSO REVE
RSE D AND L, SO D=L AND L=D
70 DATA172,68,125,38,1,205,75,127,24,163:  'TO CHANGE ROUTINE SO
 IT DOES ONLY ONE LOOP, CHANGE FIRST DATA ITEM IN THIS LINE TO 1
79 (NOW 172) AND LAST ONE IN THIS LINE TO 170 (NOW 163). CAN STI
LL USE VALUE OF 100 TO SPECIFY A LINE OR DOT.
80 DATA0,0,0,0
90 PRINT@24,"    "
100 PRINT@0,;
110 INPUT"INPUT 100 FOR DOT ONLY";A
120 X=USR(A)
130 PRINT@970,"X COORD =";INT(X/256),"Y COORD =";XAND255;
140 GOTO90
```

I hope that these simple demonstration programs will convince you that the unit is very easy to program and use. If anybody develops a good program using this board, send it to MICRO-80 for their evaluation - you may even work out a way to modify existing games programs to use the joysticks instead of the keyboard. We at the Adelaide Users' Group plan to develop other circuits which use this board as the main interface to the computer. Any that I design will probably be offered to MICRO-80 to publish if they feel it is "what the people want". We have a few ideas at the moment but any suggestions would be welcome. I may be contacted through the Users' Group address or MICRO-80 will pass on any readers' requests that they receive. I hope that those of you who construct this board get as much fun and use out of it as I have in the short time that it has been finished. I'm sure that hardware buffs will love it.

- 0000000000 -

***** THE THEORY AND TECHNIQUES OF SORTING - Part 2 by B. Simson *****

Last month we considered a simple algorithm for sorting some numbers in memory using a technique of sorting by transposition or exchange, known as the bubble sort. However, if you ran the program LISTed there for a large number of digits (say 100 numbers) then you would have realised the considerable length of time it took to sort them. This is because in a bubble sort, the time taken is related to the square of the number of items being sorted (N^2). Such a relationship means that the algorithm is not suited for large numbers and is commonly limited to sorting lists of about 15 items. We could increase the efficiency somewhat by employing a technique known as "MERGING" substrings of sorted data, along with the bubble sort.

Sorting often involves a trade-off between several resources, say memory and time. Some sorting algorithms are fast, but at the expense of using a large portion of memory in the process, depending on the application. Here, we shall consider a trade-off between actual sorting time, and time spent merging sorted substrings (subsets).

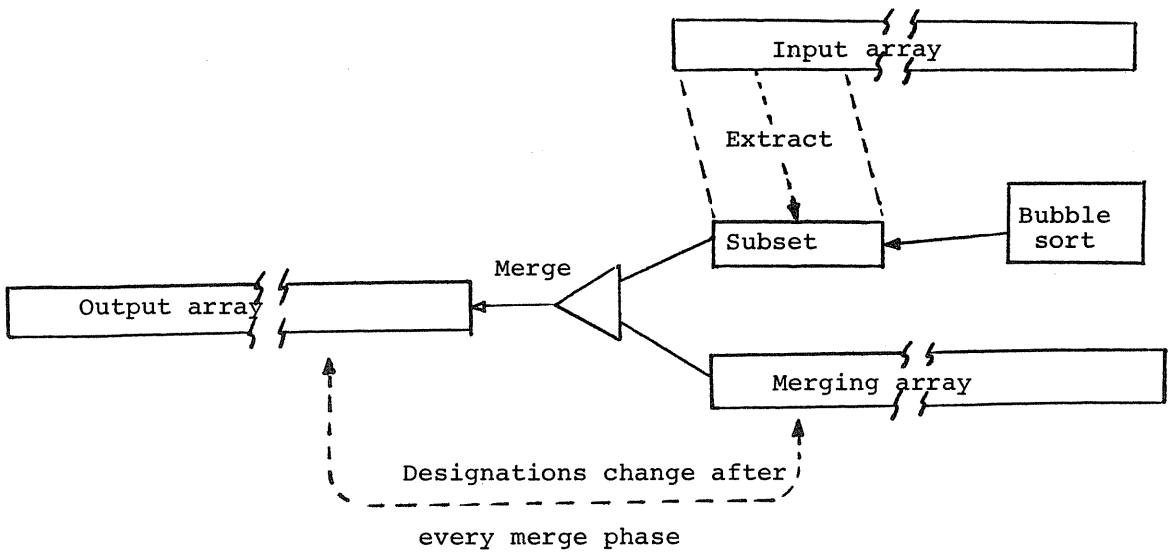


Figure 1. Merging extracted subsets of the input array.

Figure 1 explains the workings of the algorithm. We have the input array from which a subset of numbers of selected quantity is extracted serially and placed in the sort area, where the bubble sort processes them to produce a sorted list. This list is then merged with either array B or array C, depending on which one has been designated the array to be merged. Array B and C alternate in this designation, i.e. if B was the array to be merged in the last merge phase, then C will be merged with the sort area in this merge phase. The output of each merge phase is put into either B or C array, depending on which was used as output in the last merge phase. B and C also alternate in this function. So, in the first merge phase, a subset is extracted from the input array and sorted, then merged with array B (which is empty in the first phase) to produce a sorted list in array C. Next, another subset is extracted and merged with array C to produce a sorted list in array B this time. This continues until all items have been extracted from the input array, after which the array used as the output in the last merge phase is determined, it being the final list of sorted data.

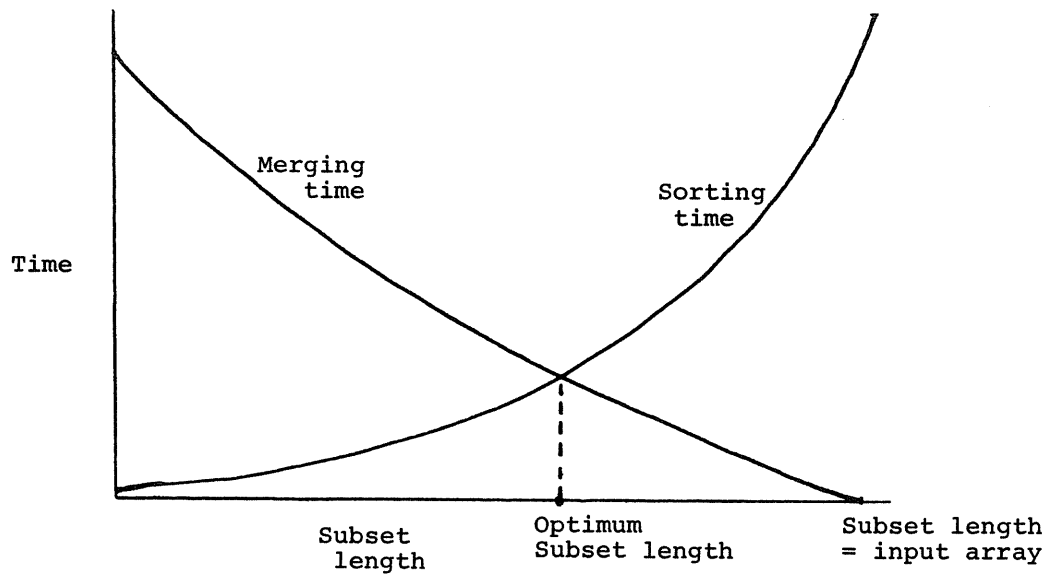


Figure 2. Optimizing merging time with sorting time.

Now it is plain that the larger the sort area, the longer it will take to actually sort (bubble), but also, the larger the sort area, the fewer extractions of subsets are made from the input array, therefore invoking fewer merge phases, and consequently less time is spent in merging. This is illustrated in Figure 2.

So a trade-off must be made, which is achieved by optimizing the subset length (therefore the sort area). The complete program employing this technique follows. An additional feature is an option to sort in descending order:

```
760 CLEAR1:CLS:DEFINTA,B,C,I:PRINT@15,"** BUBBLE SORT/MERGE **":PRINT
780 INPUT"SORT HOW MANY NO'S";N:DIMI(N),B(N),C(N)
800 RANDOM:INPUT"RANGE (-32767 TO 32767) FROM";R1:PRINTTAB(26);:INPUT"TO";R2:IFR
2<=R1GOTO800
820 FORT%=1TON:I(T%)=RND(R2-R1)+R1:NEXT:T=T%
840 SL=INT(SQR(N*2))+1:IFSL>NTHENSL=N:  REM DETERMINE OPTIMUM
SUBSET LENGTH.
860 DIMA(SL):     REM   A IS THE SORT AREA DESCRIBED ABOVE.
880 AD$="":INPUT"ASCENDING OR DESC.(A/D)";AD$:IFAD$="A"GOTO920
900 IFAD$<>"D"PRINT"  HUH ???":GOTO880
920 PRINT"...SORTING...":FC=1
940 REM  EXTRACT SUBSET FROM INPUT ARRAY
960 FORT=1TOSL:A(T)=I(FC):FC=FC+1
980 IFFC>NTHENT=T+1:GOTO1020
1000 NEXT
1020 GOSUB1740:T=N3+2
1040 ' ** MERGE ROUTINE **
1060 ' CA,CB,CC ARE ELEM COUNTERS A,B,C
1080 CA=1:CB=1:CC=1
1100 IFAD$="D"GOTO1480
1120 IFB1=1GOTO1320
1140 ' TEST FOR EOF FOR EITHER ARRAY
1160 IFCB>S1FORL=CATOT-1:C(CC)=A(L):CC=CC+1:NEXT:GOTO1240
1180 IFCA=TFORL=CBTOS1:C(CC)=B(L):CC=CC+1:NEXT:GOTO1240
1200 IFA(CA)<B(CB)C(CC)=A(CA):CA=CA+1ELSEC(CC)=B(CB):CB=CB+1
1220 CC=CC+1:GOTO1160
1240 B1=1:S1=CC-1:GOTO1380 'MERGE COMPLETE,CHANGE MERGE ARRAY,SET EOF
1260 ' ARRAY C IS MERGE ARRAY
1280 IFCC>S1FORL=CATOT-1:B(CB)=A(L):CB=CB+1:NEXT:GOTO1360
1300 IFCA=TFORL=CCTOS1:B(CB)=C(L):CB=CB+1:NEXT:GOTO1360
1320 IFA(CA)<C(CC)B(CB)=A(CA):CA=CA+1ELSEB(CB)=C(CC):CC=CC+1
1340 CB=CB+1:GOTO1280
1360 B1=0:S1=CB-1
1380 M=M+1:IFFC<=NGOTO960
1400 REM PRINT SORTED DATA
1420 FORT=1TOS1:IFB1=0PRINTB(T);ELSEPRINTC(T);
1440 NEXT:PRINT:PRINT"  SUBSET LENTH:";SL:PRINT"  NO OF MERGES:";M:END
1460 ' DESCENDING SEQ.
1480 IFB1=1GOTO1640
1500 IFCB>S1FORL=CATOT-1:C(CC)=A(L):CC=CC+1:NEXT:GOTO1580
1520 IFCA=TFORL=CBTOS1:C(CC)=B(L):CC=CC+1:NEXT:GOTO1580
1540 IFA(CA)>B(CB)C(CC)=A(CA):CA=CA+1ELSEC(CC)=B(CB):CB=CB+1
1560 CC=CC+1:GOTO1500
1580 B1=1:S1=CC-1:GOTO1380
1600 IFCC>S1FORL=CATOT-1:B(CB)=A(L):CB=CB+1:NEXT:GOTO1680
1620 IFCA=TFORL=CCTOS1:B(CB)=C(L):CB=CB+1:NEXT:GOTO1680
1640 IFA(CA)>C(CC)B(CB)=A(CA):CA=CA+1ELSEB(CB)=C(CC):CC=CC+1
1660 CB=CB+1:GOTO1600
1680 B1=0:S1=CB-1:GOTO1380
1700 '  ** SORT ROUTINE **
1720 ' T=SUBSET LEN. +1,N=LEN. OF INPUT SUBSET.
1740 T=T-1:N3=T-1:FORN1%=1TON3:N4=0
1760 FORN2%=1TO(T-1):IFAD$="D"GOTO1860
1780 IFA(N2%)<=A(N2%+1)GOTO1820
1800 A1=A(N2%):A(N2%)=A(N2%+1):A(N2%+1)=A1:N4=1
1820 NEXT:IFN4=0RETURN
1840 T=T-1:NEXT:RETURN
1860 IFA(N2%)>=A(N2%+1)GOTO1820ELSEGOTO1800
```

The program displays the length of each subset used for extraction and sorting, and the number of merges involved in the whole process.

By using merging techniques with the bubble sort, an increase in efficiency of 270% was obtained, based on an input list size of 100 items. That is:-

100 items with straight bubble .... 130 seconds
100 items with bubble sort/merge ... 48 seconds.

It can be seen that merging techniques strip a considerable amount of time off the entire process. In fact, merging is such an efficient technique that an entire sorting algorithm, known as the merge sort, is based on merging principles.

We shall leave the bubble sort algorithm here, and consider some other sorting techniques of varying efficiencies. Last month I finished off by saying that the next article will consider techniques of sorting "by insertion". However, this will be postponed and instead, another simple sorting routine will be presented, this time involving a technique of sorting "by selection". This algorithm involves a search of elements in an array to locate the one with the smallest value (or "key"). When found, it is exchanged with the first element in the array. This places the element with the smallest key in the first position in the array. This process is repeated for the remaining elements to select what will become the second smallest value which is interchanged with the second element in the array. The process of searching for the element with the next smallest value and placing it in its proper position (by exchanging elements) continues until all elements have been sorted. This is known as the selection sort, the program for which appears below:

```
2110 REM          N=NO OF ELEMENTS TO BE SORTED
2120 REM          MIN=MINIMUM FOUND SO FAR
2130 REM          P=POINTER TO MIN
2140 REM          I=PASS COUNTER
2150 REM          T=TEMPORARY VARIABLE USED FOR EXCHANGING
2160 REM
2170 IF N<2 THEN 2250
2180 FOR I=1 TO (N-1)
2190     MIN=A(I):P=I     'SET MINIMUM BEFORE SEARCH
2200     FOR J=(I+1) TO N          'START FROM WHERE YOU ARE
2210         IF A(J)<MIN THEN MIN=A(J):P=J    'FOUND A LOWER ONE
2220     NEXT J
2230     IF P<>I THEN T=A(I):A(I)=A(P):A(P)=T     'EXCHANGE
2240 NEXT I
2250 RETURN
2260     <END OF PROGRAM>
2270 '
2280 'HERE FOLLOWS THE DRIVER, WHICH BUILDS THE INPUT ARRAY,
2290 'DISPLAYS THE DATA, CALLS THE SORT, AND DISPLAYS THE
2300 'SORTED RESULTS.
```

Here follows the driver, which builds the input array, displays the data, calls the sort, and displays the sorted results.

```
2320 INPUT"HOW MANY NUMBERS";N
2330 CLS: DIM A(N): RANDOM
2340 FOR I=1 TO N
2350     A(I)=RND(1000):PRINTA(I);
2360 NEXT I: PRINT
2370 INPUT"HIT ENTER TO START";I
2380 GOSUB 2170
2390 FOR I=1TO N
2400     PRINTA(I);
2410 NEXT I
2420 END
```

To further explain the functions of this algorithm, consider the trace of the first pass.

Assume array before sort is: 12, 18, 15, 6, 13.

MIN	CONSIDERING...	POSITION OF MIN
12	18	1
12	15	1
12	6 (Moved to min)	4 (updated)
6	13	4

Now, since the position of the minimum has changed from that at the start of the pass, an exchange is performed between the 1st and 4th items, so that the data after the first pass looks like this:

6, 18, 15, 12, 13.

The second pass will commence searching data from the second item onwards (i.e. 18) since position 1 is occupied by the smallest value in the array, and hence is its final position.

The efficiency of the selection sort is also of order (N^2). A later article will examine the efficiency and compare it, by mathematical analysis, with other sorting routines.

This means that the selection sort is also limited to small (15-20) input table sizes to obtain any reasonable response times. However, it may be more efficient than the bubble sort if implemented, using the machine's instruction set, on a machine having firmware available to find the smallest element in a list at high speed.

TO SUMMARIZE...

It was discovered that sorting routines like the bubble sort are not efficient for large list sizes. The time taken is proportional to N^2. The efficiency of such sorts can be improved somewhat by employing the technique of merging, which involves a trade-off of some kind. Another simple sorting routine is available, based on the principle of straight selection and known as the selection sort.

Next month we shall be looking at yet another technique of sorting, that of sorting "by insertion" along with some of its variations.

- 0000000000 -

**** SOFTWARE SECTION ****

***** TIC TAC TOE - Level I TRS-80 by B. Smith *****

If you are like me, the first computer game I ever wanted to play on owning a computer was TIC TAC TOE (noughts and crosses). So I purchased a Tandy's game pack that had a program called random TIC TAC TOE. This game to me was a let-down because firstly, it was slow and secondly, at every input it altered the location of the 'X's and 'O's around the board. Very confusing!

So I developed this game which is fast and 'thinks' its next move. Remember, it will win first before blocking your move. It should keep you on your toes. The computer always plays the noughts and the human opponent the X's.

Lines 6-11	draw the grid and number the squares.
Lines 13-25Ø	allow the human player to choose the position for his next X, checks input for validity then places the 'X' in a valid position.
Lines 26Ø-274	redraw previous positions of 'X' and 'O' after every new input.
Lines 28Ø-295	computer's move.
Lines 399-418	test for the presence of a winning combination.
Lines 58Ø-622	test for two X's in a line so that computer may block them.
Lines 625-	check if there are two 'O's on any line and places a third to give victory to the computer.

```
 1  REM * TIC TAC TOE FOR LEVEL 1 16K RAM USERS *
 2  REM * BY B SMITH 7 INNES ST ALBANY WA *
 3  CLS:P.A.463,"DO YOU WANT TO GO FIRST.(YES OR NO)";:I.M
 4  A=0:B=0:C=0:D=0:E=0:F=0:G=0:H=0:I=0:Q=0:L=0:Z=0
 5  CLS:P.A.27,"TIC TAC TOE"
 6  P.A.84,"1":P.A.93,"2":P.A.102,"3":P.A.276,"4":P.A.285,"5"
 7  P.A.294,"6":P.A.468,"7":P.A.477,"8":P.A.486,"9"
 8  X=56:F.Y=3TO28:S.(X,Y):N.Y
 9  X=73:F.Y=3TO28:S.(X,Y):N.Y
 10  Y=11:F.X=40TO89:S.(X,Y):N.X
 11  Y=20:F.X=40TO89:S.(X,Y):N.X
 12  IFM=NT.280
 13  P.A.704,"YOUR MOVE (X) - WHICH NO.";:I.J:IF(J>9)+(J<1)T.250
 14  Z=1:IFJ=1T.23
 15  IFJ=2T.25
 16  IFJ=3T.27
 17  IFJ=4T.29
 18  IFJ=5T.31
 19  IFJ=6T.33
 20  IFJ=7T.35
 21  IFJ=8T.37
 22  IFJ=9T.39
 23  IFA>0T.240
 24  A=1:P.A.152,"X":G.260
 25  IFB>0T.240
```

```
 26  B=1:P.A.160,"X":G.260
 27  IFC>0T.240
 28  C=1:P.A.168,"X":G.260
 29  IFD>0T.240
 30  D=1:P.A.344,"X":G.260
 31  IFE>0T.240
 32  E=1:P.A.352,"X":G.260
 33  IFF>0T.240
 34  F=1:P.A.360,"X":G.260
 35  IFG>0T.240
 36  G=1:P.A.536,"X":G.260
 37  IFH>0T.240
 38  H=1:P.A.544,"X":G.260
 39  IFI>0T.240
 40  I=1:P.A.552,"X":G.260
240   P.A.704,"YOU CAN'T USE THIS NO..IT'S ALREADY IN USE. TRY AGAIN."
241  F.N=1T01500:N.N:P.A.704,"                                        "
242  G.13
250  P.A.704,"YOU MUST USE NUMBERS BETWEEN 1 & 9.":G.241
260  IFB=1T.P.A.160,"X"
261  IFB=4T.P.A.160,"O"
262  IFC=1T.P.A.168,"X"
263  IFC=4T.P.A.168,"O"
264  IFE=1T.P.A.352,"X"
265  IFE=4T.P.A.352,"O"
266  IFF=1T.P.A.360,"X"
267  IFF=4T.P.A.360,"O"
268  IFH=1T.P.A.544,"X"
269  IFH=4T.P.A.544,"O"
270 IFI=1T.P.A.552,"X"
271 IFI=4T.P.A.552,"O"
272  GOS.399
273  IFZ=2T.280
274 G.13
280  P.A.704," IT'S MY TURN.........":F.N=1T01000:N.N
282  IFQ=0T.GOS.550:G.284
283  IFQ=1T.625
284  IFL=1T.P.A.152,"O":A=4:G.260
285  IFL=2T.P.A.160,"O":B=4:G.260
286  IFL=3T.P.A.168,"O":C=4:G.260
287  IFL=4T.P.A.344,"O":D=4:G.260
288  IFL=5T.P.A.352,"O":E=4:G.260
289  IFL=6T.P.A.360,"O":F=4:G.260
290  IFL=7T.P.A.536,"O":G=4:G.260
291  IFL=8T.P.A.544,"O":H=4:G.260
292  IFL=9T.P.A.552,"O":I=4:G.260
293  GOS.399
294  Z=Z-2
295  G.13
398  E.
399  X=56:F.Y=3T028:S.(X,Y):N.Y:X=73:F.Y=3T028:S.(X,Y):N.Y
400  K=A+B+C:IFK=3T.500
401  IFK=12T.505
402  K=D+E+F:IFK=3T.500
403  IFK=12T.505
404  K=G+H+I:IFK=3T.500
405  IFK=12T.505
406  K=A+D+G:IFK=3T.500
407  IFK=12T.505
408  K=B+E+H:IFK=3T.500
409  IFK=12T.505
410  K=C+F+I:IFK=3T.500
411  IFK=12T.505
412  K=A+E+I:IFK=3T.500
413  IFK=12T.505
414  K=C+E+G:IFK=3T.500
415  IFK=12T.505
416  IF(A>0)*(B>0)*(C>0)*(D>0)*(E>0)*(F>0)*(G>0)*(H>0)*(I>0)T.510
417  Z=Z+1
418  RET.
500  P.A.704,"                                                        "
501  P.A.714,"***  Y O U  W I N  -  W E L L  D O N E.  ***"
502  P.A.849,"WANT TO PLAY AGAIN.(YES OR NO)";:I.M
503  IFM=YT.3
504  CLS:E.
505  P.A.704,"                                          "
```

```
 506  P.A.716,"***  I  B E A T  Y O U  D U M M Y.  ***"
 507  G.502
 510  P.A.704,"                                    "
 511  P.A.720,"*** I T ' S  A  D R A W.  ***"
 512  G.502
 550  Q=1:L=RND(9)
 551  IFL=1T.561
 552  IFL=2T.563
 553  IFL=3T.565
 554  IFL=4T.567
 555  IFL=5T.569
 556  IFL=6T.571
 557  IFL=7T.573
 558  IFL=8T.575
 559  IFL=9T.577
 560  RET.
 561  IFA>0T.550
 562   RET.
 563  IFB>0T.550
 564  RET.
 565  IFC>0T.550
 566  RET.
 567  IFD>0T.550
 568  RET.
 569  IFE>0T.550
 570  RET.
 571  IFF>0T.550
 572  RET.
 573  IFG>0T.550
 574  RET.
 575  IFH>0T.550
 576  RET.
 577  IFI>0T.550
 578  RET.
 580  REM * TEST FOR POSITION OF 'X' TO BLOCK *
 581  IF((A=1)*(B=1))+((A=1)*(C=1))+((C=1)*(B=1))T.591
 582  IF((D=1)*(E=1))+((D=1)*(F=1))+((E=1)*(F=1))T.595
 583  IF((G=1)*(H=1))+((G=1)*(I=1))+((H=1)*(I=1))T.599
 584  IF((A=1)*(D=1))+((A=1)*(G=1))+((D=1)*(G=1))T.603
 585  IF((B=1)*(E=1))+((E=1)*(H=1))+((B=1)*(H=1))T.607
 586  IF((C=1)*(F=1))+((F=1)*(I=1))+((C=1)*(I=1))T.611
 587  IF((A=1)*(E=1))+((E=1)*(I=1))+((A=1)*(I=1))T.615
 588  IF((C=1)*(E=1))+((E=1)*(G=1))+((C=1)*(G=1))T.619
 589  Q=0:G.282
 591  IF(A>0)*(B>0)*(C>0)T.582
 592  IFA=0T.L=1:G.284
 593  IFB=0T.L=2:G.285
 594  L=3:G.286
 595  IF(D>0)*(E>0)*(F>0)T.583
 596  IFD=0T.L=4:G.287
 597  IFE=0T.L=5:G.288
 598  L=6:G.289
 599  IF(G>0)*(H>0)*(I>0)T.584
 600  IFG=0T.L=7:G.290
 601  IFH=0T.L=8:G.291
 602  L=9:G.292
 603  IF(A>0)*(D>0)*(G>0)T.585
 604  IFA=0T.L=1:G.284
 605  IFD=0T.L=4:G.287
 606  L=7:G.290
 607  IF(B>0)*(E>0)*(H>0)T.586
 608  IFB=0T.L=2:G.285
 609  IFE=0T.L=5:G.288
 610  L=8:G.291
 611  IF(C>0)*(F>0)*(I>0)T.587
 612  IFC=0T.L=3:G.286
 613  IFF=0T.L=6:G.289
 614  L=9:G.292
 615  IF(A>0)*(E>0)*(I>0)T.588
 616  IFA=0T.L=1:G.284
 617  IFE=0T.L=5:G.288
 618  L=9:G.292
 619  IF(C>0)*(E>0)*(G>0)T.589
 620  IFC=0T.L=3:G.286
 621  IFE=0T.L=5:G.288
 622  L=7:G.290
```

```
624  REM * TEST FOR POSITION OF 'O' TO WIN *
625  IF((A=4)*(B=4))+((A=4)*(C=4))+((B=4)*(C=4))T.591
626  IF((D=4)*(E=4))+((D=4)*(F=4))+((E=4)*(F=4))T.595
627  IF((G=4)*(H=4))+((G=4)*(I=4))+((H=4)*(I=4))T.599
628  IF((A=4)*(D=4))+((A=4)*(G=4))+((D=4)*(G=4))T.603
629  IF((B=4)*(E=4))+((E=4)*(H=4))+((B=4)*(H=4))T.607
630  IF((C=4)*(F=4))+((F=4)*(I=4))+((C=4)*(I=4))T.611
631  IF((A=4)*(E=4))+((E=4)*(I=4))+((A=4)*(I=4))T.615
632  IF((C=4)*(E=4))+((E=4)*(G=4))+((C=4)*(G=4))T.619
633  G.581
```

***** ATTACK Disk Basic - L2/16K by M. Bloss *****

Attack is a fast, exciting game with sound. Your space ship is at the bottom of the screen but all that is shown is your cannon. An alien ship appears at the top of the screen and fires rockets at you. You must intercept each rocket by firing your cannon and blowing it up.

The closer the rocket is to the alien ship when you hit it, the higher your score. There are eight columns on the screen down which the alien fires, and in which the cannon is able to move and fire. If you miss a rocket it will destroy you which terminates the game. The longer you play, the faster the game.

To hear the sound, connect an audio amplifier to the plug that normally goes into the AUX socket of the CTR-80. Alternatively, remove all the plugs from your cassette recorder, connect the AUX plug into the MIC socket on the recorder, press down the PLAY key and turn up the volume. Thanks are due to C.E. Kendall for his KEYBOARD BLEEPER program published in MICRO-80, Issue 12.

When entering the program, Disk users should remove the apostrophe (') from the beginning of line 40. Level II users may omit line 40 altogether.

```
20 '

     SYSTEM-80 OWNERS ADJUST LINES 1080 TO 1100 AS PER COMMENTS



30 GOSUB1050
40 CLEAR1500:DIMNA$(30):ZZ=50:'


50 'DEFUSR=&H407B  :REMEMBER  DISK USERS UNCOMMENT THIS LINE


60 GOSUB360
70 SC=0:SH=0:G=G+1:CLS:PRINT@532,NA$(G);"'S TURN":FORGG=1TO2500:
NEXTGG:CLS
80 GOSUB330
90 X=16352
100 M=66:GOSUB820
110 GOSUB140
120 RANDOM
130 GOTO210
140 A$=INKEY$
150 IFA$="1"THENX=X-5:IFX<16339THENX=X+5
160 IFA$="1"THENGOSUB840
170 IFA$="2"THENX=X+5:IFX>16380THENX=X-5
180 IFA$="2"THENGOSUB1010
190 IFA$=" "THENGOSUB290
200 GOSUB280:RETURN
210 A=RND(8):FORZ=1TOZZ:GOSUB140:NEXT:ZZ=ZZ-2
220 A=5*A+17
230 R=USR(-5120):FORAA=A+15360TOA+16256STEP64
240 GOSUB140:POKEAA,144:POKEA+15359,130:POKEA+15360,134:IFAA>154
24THENPOKEAA-64,128
250 POKEAA,128
260 NEXTAA
270 GOTO710
```

```
280 POKEX-1,160:POKEX,180:RETURN
290 FORXX=X-64TOX-960STEP-64:POKEXX,132:POKEXX+64,128:GOSUB280:P
OKEXX,128:IFXX=AATHENC=1:XX=X-960
300 NEXTXX:IFC=1THEN740
310 POKEX-960,128
320 RETURN
330 FORN=19TO60STEP41:FORO=15360+NTO16320+NSTEP64:POKEO,149:NEXT
O,N
340 XN=77:FORNN=600TOOSTEP-100:PRINT@XN,NN;:IFNN=100THENXN=XN+13
OELSEXN=XN+128
350 NEXT:RETURN
360 CLS:FORL=1TO7:PRINTCHR$(23);:PRINT"A T T A C K ! !":PRINT:NE
XTL
370 FORL=1TO50:A=USR(L*256):FORLL=1TO25:NEXTLL,L
380 CLS:PRINT"YOU ARE ON THE SPACESHIP ,THE GALACTIC RUN"
390 PRINT"WHILE PASSING THE GALAXY 74RV2,YOU COME UNDER ATTACK"
400 PRINT"BY A FLEET OF ALIENS."
410 PRINT"YOUR JOB IS TO SHOOT DOWN THESE ALIENS AND SCORE POINT
S."
420 PRINT:PRINT"YOUR SHIP IS AT THE BOTTOM OF THE SCREEN AND ONL
Y THE"
430 PRINT"GUN IS SHOWN.THE ALIENS FIRE FROM THE TOP OF THE"
440 PRINT"SCREEN DOWN ANY OF THE 8 COLUMNS COVERED BY YOUR SPACE
SHIP"
450 PRINT"YOU HAVE TO INTERCEPT THE MISSILE AND BLOW IT UP BEFOR
E"
460 PRINT"IT REACHES YOUR SPACESHIP AND DESTROYS IT."
470 GOSUB650
480 CLS:PRINT"TO MOVE YOUR GUN PRESS THE <1> KEY TO MOVE LEFT AN
D"
490 PRINT"THE <2> KEY TO MOVE RIGHT.TO FIRE PRESS THE <SPACE> BA
R"
500 PRINT"THE AMOUNT OF POINTS YOU GET DEPENDS ON HOW FAR UP"
510 PRINT"THE SCREEN YOU HIT THE MISSILE."
520 PRINT"THE NUMBERS TO THE LEFT"
530 PRINT"TELL YOU HOW MUCH YOU GET."
540 PRINT"IF YOU GET HIT THAT IS THE END OF YOUR GAME."
550 PRINT"AS THE GAME PROGRESSES,THE ALIENS FIRE AT YOU MORE AGR
ESSIVELY."
560 PRINT"TO WIN YOU MUST HAVE THE HIGHEST SCORE"
570 PRINT:INPUT"HOW MANY PLAYERS(MINIMUM OF 2)";P
580 IFP<2THEN570
590 CLS:PRINT"ENTER YOUR NAMES ONE AT A TIME"
600 FORI=1TOP:INPUTNA$(I):NEXTI:RETURN
610 SC(G)=SC:SH(G)=SH:PRINT"END OF GO":IFSH=20THENPRINT"YOU HAVE
 RUN OUT OF ROCKETS"
620 IFG=PTHEN860
630 PRINT"NEXT PLAYER PLEASE PRESS ENTER TO START"
640 INPUTR$:GOTO70
650 PRINT"PRESS ANY KEY TO CONTINUE"
660 FORL=1TO10000:Q$=INKEY$:IFQ$<>""THENL=10000:GOTO680
670 LL=RND(50)+70:R=USR(LL*256):FORLA=1TO25:NEXTLA
680 NEXTL:RETURN
690 RETURN
700 GOTO700
710 GOSUB970:FORV=1TO40:R=USR(-2560):NEXTV:CLS:PRINT"K A B O O M
 ! !"
720 PRINT"A MISSILE HIT YOUR SHIP AND BLEW IT UP"
730 GOTO610
740 C=0:POKEXX,128:POKEXX-65,128:POKEXX-64,128:GOSUB1030:FORV=1T
012:R=USR(0):NEXTV:GOSUB1040:FORQQ=1TO100:NEXTQQ:IFAA<15551THENSC
=SC+600:GOTO830
750 IFAA<15679THENSC=SC+500:GOTO830
760 IFAA<15807THENSC=SC+400:GOTO830
770 IFAA<15935THENSC=SC+300:GOTO830
780 IFAA<16063THENSC=SC+200:GOTO830
790 IFAA<16191THENSC=SC+100
800 GOTO830
810 IFSC<10000THENM=65ELSEM=64
820 PRINT@1,"SCORE";:PRINT@M,SC;:RETURN
830 GOSUB810:FORER=A-1TO895+ASTEP64:PRINT@ER,STRING$(2,128);:NEX
TER:GOTO210
840 IFX<16339THENRETURN
850 POKEX+4,128:POKEX+5,128:RETURN
860 PRINT"THESE ARE THE FINAL SCORES"
870 FORG=1TOP
```

```
880 PRINTNA$(G);" USED UP";SH(G);"SHOTS AND SCORED";SC(G)
890 FORS=1TO500:NEXTS,G
900 FORG=1TOP
910 FORD=1TOP
920 IFSC(G)<=SC(D)THEN940
930 WW(G)=WW(G)+1:IFWW(G)=P-1THEN950
940 NEXTD,G
950 PRINT:PRINTNA$(G);" WON WITH";SC(G);"POINTS.CONGRATULATIONS!
!"
960 PRINT:INPUT"PLAY AGAIN";A$:A$=LEFT$(A$,1):IFA$="Y"THEN40ELSE
PRINT"GOODBYE FOR NOW":END
970 POKEX-1,128:POKEX,128:POKE16342,161:POKE16346,181:POKE16353,
177:POKE16359,179:POKE16365,178:POKE16372,186:POKE16376,146
980 POKE16276,161:POKE16281,161:POKE16288,161:POKE16294,177:POKE
16296,178:POKE16302,146:POKE16309,146:POKE16314,146
990 POKE16216,162:POKE16223,162:POKE16229,161:POKE16233,146:POKE
16239,145:POKE16246,145
1000 POKE16158,161:POKE16164,162:POKE16170,145:POKE16176,146:POK
E16100,145:POKE16106,162:PRINT@483,"K A B O O M !";:RETURN
1010 IFX>16380THENRETURN
1020 POKEX-6,128:POKEX-5,128:RETURN
1030 POKEAA+63,146:POKEAA+65,161:POKEAA-65,161:POKEAA-63,146:POK
EAA-1,132:POKEAA-2,132:POKEAA+1,136:POKEAA+2,136:RETURN
1040 POKEAA-2,128:POKEAA+2,128:FORWS=AA-63TOAA+65STEP64:POKEWS,1
28:NEXT:FORWS=AA-65TOAA+63STEP64:POKEWS,128:NEXT:RETURN
1050 FORJ=16435TO16437:READI:POKEJ,I:NEXT
1060 FORJ=16480TO16513:READI:POKEJ,I:NEXT
1070 POKE16405,0
1080 DATA195,96,64,205,227,3,183,200,8,14,20,68,62,1,211,255,16,
254,68,62,2,211,255,16,254,13,32,239,8,201,205,127,10,205,101,64,
201
1090 REMEMBER TO UNCOMMENT THIS LINE AND DELETE LINE 1070 IF
YOUR COMPUTER IS A SYSTEM 80. DELETE ALL OF THIS LINE UP TO
HERE. DATA195,96,64,205,227,3,183,200,8,14,20,68,62,5,211,255,16
,254,68,62,6,211,255,16,254,13,32,239,8,201,205,127,10,205,101,64
,201
1100 REMEMBER TO UNCOMMENT THIS LINE AS WELL, UP TO HERE.
OUT254,255
1110 POKE16526,123:POKE16527,64:RETURN
```

***** BASIC LINE REFERENCE VALIDATOR Level 2/4K-16K by K. Shillito *****

This little program snuggles itself into a part of memory that the BASIC interpreter never uses. Thus, even if the memory is "full", this program can still be loaded and run! You can create it using BMON or with an assembler.

** PROGRAM FUNCTION **

The purpose of this program is to detect all instances of reference to non-existent lines in BASIC programs, (including direct commands, except the LIST command, which does not generate an error for non-existent lines).

** LOADING AND RUNNING THE PROGRAM **

Type SYSTEM, then NONAME, then press BREAK.
The program is now available whenever required. To run it, press SYSTEM, followed by /16455.
The program will give ?FC ERROR followed by the number of the first line it encounters containing a reference to a non-existent line number (e.g. GOTO 3ØØ, where 3ØØ doesn't exist). After correcting the error, run it again to see if there are further errors. It is advisable to run this program before using a RENUMBER utility (such as BMON).

** WARNING **

This program may give spurious error messages if you have somehow POKEd graphics characters into strings or remarks. This is not normally possible unless you used a special utility such as S-KEY to do it.

** TECHNICAL COMMENTS **

This program occupies only 59 bytes and resides from 16455 to 16504, an area of memory that is never used by BASIC. Hence, it can be used even if memory is completely full. The program will not detect references unaccompanied by key words. Also, it will detect certain types of syntax errors. It does not detect a reference in the form 'DELETE-3Ø, which is unlikely to occur anyway. Apart from these exceptions, it should pick up all references, including some which BMON's RENUMBER misses.

** RELOCATION **

The program can be relocated. No re-assembly is needed.

** HEX CODE **

```
4047-404E  2A A4 40 D7 CA 66 00 23
404F-4056  4E 23 46 ED 43 A2 40 D7   START ADDRESS 4047
4057-405E  FE 00 28 1E EB 21 7C 40   NAME  NONAME
405F-4066  01 07 00 ED B1 EB 20 EF
4067-406E  D7 30 ED CD 5A 1E E5 CD
406F-4076  2C 1B E1 D2 4A 1E AF BE
4077-407E  20 EE 23 18 CE 91 8D 95
407F-4082  CA 8E 9F B6
```

** METHOD OF OPERATION **

The program uses an area of RAM starting at 4Ø47H and ending with the USR pointers that is never used by BASIC. It makes the following assumptions about BASIC syntax:

1. Basic line references only occur after the tokens (in hex) 8D, 91, CA, 95, 8E or B6 (the LIST tokens B4 and B5 cannot have invalid line references, so they are not included in the program), possibly with interspersed blanks.

2. Where a line reference occurs, followed by a non-zero byte that is an ASCII digit, then that digit must begin another line reference, e.g.

```
GOSUB 1Ø,2Ø
DELETE 2-4Ø
```

   (A digit should not follow a colon).

3. The tokens in Note 1 above occur only as such (hence, the program must not contain graphic characters in string constants or REM's.

(Actually, Note 1 is not quite true - this is why statements in the form DELETE -2Ø or IF(X)2Ø are overlooked).

The mode of operation of each section of the EDTASM source code is explained below.

PROCESS LINE POINTERS AND LINE NUMBER

This returns to BASIC at the end of the program. (I find 66H to be better than 6CCH, since the former repairs the stack). If the end is not yet reached, then it places the line being processed in the pointers in case an error message is to be generated.

SEEK TOKENS IN LINE

This looks at each byte in a line, going to the section above, if the EOL token is encountered, or to the section below if one of the tokens in Note 1 above is encountered.

SEEK LINE REF. AFTER TOKEN

This returns to the previous section if a token (or comma or dash as per (2) above), is not followed by a digit.

VALIDATE LINE REFS.

This uses ROM routines to convert the line ref. from ASCII to hex, and then to check that it exists. If not, then FC ERROR is generated, since I don't know the entry point for UL ERROR. (it is 1ED9 - Ed.)

```
                  00100 ;********************
                  00110 ;*BASIC LINE REFERENCE VALIDATION ROUTINE
                  00120 ;*KEN SHILLITO          DECEMBER 1980
                  00130 ;********************
                  00140 ;PROCESS LINE POINTERS AND LINE NUMBER
4047              00150          ORG     4047H            ;UNUSED RESERVED MEM
4047 2AA440       00160 VALID    LD      HL,(40A4H)       ;(HL)=START BASIC
404A D7           00170 NEXLIN   RST     10H              ;(HL)=A=MSB LINE PTR
404B CA6600       00180          JP      Z,66H            ;RESET AT END
404E 23           00190          INC     HL               ;(HL)=LSB LINE PTR
404F 4E           00200          LD      C,(HL)
4050 23           00210          INC     HL               ;(HL)=MSB LINE NO.
4051 46           00220          LD      B,(HL)           ;BC=LINE NUMBER
4052 ED43A240     00230          LD      (40A2H),BC       ;IN CASE OF ERROR
```

DON'T BE HELD BACK BY AN ANTIQUATED DISK OPERATING SYSTEM

MOVE UP TO

NEWDOS 80 **$149 incl. p&p**

NEWDOS 80 is a completely new DOS for the TRS-80 SYSTEM 80. It is well-documented, bug free and increases the power of your system many times over. It is upward compatible with TRSDOS AND NEWDOS (ie TRSDOS and NEWDOS+ programs will run on NEWDOS 80 but the reverse is not necessarily so).

These are just a few of the many new features offered by NEWDOS 80.

* New BASIC commands that support variable record lengths up to 4095 bytes long.
* Mix or match disk drives. Supports any track count from 18 to 96. Use 35, 40, 77 or 80 track 5¼ inch mini disk drives, 8 inch disk drives OR ANY COMBINATION.
* An optional security boot-up for BASIC or machine code application programs. User never sees "DOS-READY" or "READY" and is unable to "BREAK", clear screen or issue any direct BASIC statements, including "LIST".
* New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
* Enhanced and improved RENUMBER that allows relocation of subroutines.
* Create powerful chain command files which will control the operation of your system.
* Device handling for routing to display and printer simultaneously.
* MINIDOS — striking the D, F and G keys simultaneously calls up a MINIDOS which allows you to perform many of the DOS commands without disturbing the resident program.
* Includes Superzap 3.0 which enables you to display/print/modify any byte in memory or on disk.
* Also includes the following utilities:
  - Disk Editor/Assembler
  - Disassembler (Z80 machine code)
  - LM offset — allows transfers of any system tape to Disk file — automatically relocated.
  - LEVEL I — Lets you convert your computer back to Level 1.
  - LVIDKSL — Saves and loads Level 1 programs to disk.
  - DIRCHECK — Tests disk directories for errors and lists them.
  - ASPOOL — An automatic spooler which routes a disk file to the printer whilst the computer continues to operate on other programs.
  - LCDVR — a lower case drives which display lower case on the screen if you have fitted a simple lower case modification.

DISK DRIVE USERS ELIMINATE CRC ERRORS AND TRACK LOCKED OUT MESSAGES FIT A PERCOM DATA SEPARATOR

$37.00 plus $1.20 p&p.

When Tandy designed the TRS-80 expansion interface, they did not include a data separator in the disk-controller circuitry, despite the I.C. manufacturer's recommendations to do so. The result is that many disk drive owners suffer a lot of Disk I/O errors. The answer is a data separator. This unit fits inside your expansion interface. It is supplied with full instructions and is a must for the serious disk user.

MPI DISK DRIVES HIGHER PERFORMANCE — LOWER PRICE

MPI is the second largest manufacturer of disk drives in the world. MPI drives use the same form of head control as 8" drives and consequently, they have the fastest track-to-track access time available — 5msec! All MPI drives are capable of single or double-density operation. Double-density operation requires the installation of a PERCOM doubler board in the expansion interface.

As well as single head drives, MPI also makes dual-head drives. A dual-head drive is almost as versatile as two single-head drives but is much cheaper.

Our MPI drives are supplied bare or in a metal cabinet — set up to operate with your TRS-80 or SYSTEM 80. All drives are sold with a 90 day warranty and service is available through MICRO-80 PRODUCTS.

MPI B51 40 Track Single Head Drive. . . . . .only $349
MPI B52 40 Track Double Head Drive. . . . .only $449

Prices are for bare drives and include p&p. Add $10.00 per drive for a cabinet and $60.00 for a power supply to suit two drives. 40 track drives are entirely compatible with 35 track drives. A 40 track DOS such as NEWDOS 80 is necessary to utilise the extra 5 tracks.

OVER 800 KILOBYTES ON ONE DISKETTE! WITH MPI 80 TRACK DRIVES

MPI 80 track drives are now available. The B91 80 track single-head drive stores 204 Kilobytes of formatted data on one side of a 5¼ inch diskette in single-density mode. In double-density mode it stores 408 Kilobytes and loads/saves data twice as quickly.

The B92 80 track dual-head drive stores 204 Kilobytes of formatted data on EACH side of a 5¼ inch diskette in single-density mode. That's 408 Kilobytes per diskette. In double-density mode, the B92 stores a mammoth 408 Kilobytes per side or 816 Kilobytes of formatted data per diskette. With two B92's and a PERCOM double, you could have over 1.6 Megabytes of on line storage for your TRS-80 for less than $1500!!

MPI B91 80 Track Single Head Drive. . . . . .only $499
MPI B92 80 Track Dual Head Drive . . . . . .only $619

Prices are for bare drives and include p&p. Add $10.00 per drive for a cabinet and $60.00 for a power supply to suit two drives. Note: 80 track drives will not read diskettes written on a 35 or 40 track drive. If drives with different track counts are to be operated on the same system, NEWDOS 80 must be used.

CARE FOR YOUR DISK DRIVES? THEN USE 3M's DISK DRIVE HEAD CLEANING DISKETTES

$30.20 incl. p&p.

Disk drives are expensive and so are diskettes. As with any magnetic recording device, a disk drive works better and lasts longer if the head is cleaned regularly. In the past, the problem has been, how do you clean the head without pulling the mechanism apart and running the risk of damaging delicate parts. 3M's have come to our rescue with SCOTCH BRAND, non-abrasive, head cleaning diskettes which thoroughly clean the head in seconds. The cleaning action is less abrasive than an ordinary diskette and no residue is left behind. Each kit contains:

- 2 head cleaning diskettes
- 1 bottle of cleaning fluid
- 1 bottle dispenser cap

PLEASE USE ORDER FORM ON PAGE 36

USE TANDY PERIPHERALS ON YOUR SYSTEM-80 VIA SYSPAND-80 - $97.50 incl. p&p

The SYSTEM-80 hardware is not compatible with the TRS-80 in two important areas. The printer port is addressed differently and the expansion bus is entirely different. This means that SYSTEM-80 owners are denied the wealth of economical, high performance peripherals which have been developed for the TRS-80. Until now, that is. MICRO-80 has developed the SYSPAND-80 adaptor to overcome this problem. A completely self-contained unit in a small cabinet which matches the colour scheme of your computer, it connects to the 50-way expansion part on the rear of your SYSTEM 80 and generates the FULL Tandy 40 way bus as well as providing a Centronics parallel printer port. SYSPAND-80 enables you to run an Exatron Stringy Floppy from your SYSTEM 80, or an LNW Research expansion interface or any other desirable peripherals designed to interface to the TRS-80 expansion port. Make your SYSTEM 80 hardware compatible with the TRS-80 via SYSPAND-80.

PROGRAMS BY MICROSOFT

EDITOR ASSEMBLER PLUS (L2/16K) $37.50 + $1.20 p&p

A much improved editor-assembler and debug/monitor for L2/16K TRS-80 or SYSTEM 80. Assembles directly into memory, supports macros and conditional assembly, includes new commands-substitute, move, copy and extend.

LEVEL III BASIC $59.95 plus $1.20 p&p

Loads on top of Level II BASIC and gives advanced graphics, automatic renumbering, single stroke instructions (shift-key entries) keyboard debounce, suitable for L2/16K and up (Not Disk BASIC)

ADVENTURE ON DISK $35.95 plus $1.20 p&p

This is the original ADVENTURE game adapted for the TRS-80. The game fills an entire diskette. Endless variety and challenge as you seek to rise to the level of Grand Master. Until you gain skill, there are whole areas of the cave that you cannot enter. (Requires 32K One Disk)

BASIC COMPILER $208 plus $2.00 p&p

New improved version, the Basic Compiler converts Disk BASIC programs to machine code, automatically. A compiled program runs, on average, 3-10 times faster than the original BASIC program and is much more difficult to pirate.

UPGRADE TO 16K FOR ONLY $30.00!!

MICRO-80's 16K MEMORY EXPANSION KIT HAS BEEN REDUCED IN PRICE EVEN MORE

Larger volume means we buy better and we pass the savings on to you. These are our proven, prime, branded 200 ns (yes, 200 nanosecond) chips. You will pay much more elsewhere for slow, 350 ns. chips. Ours are guaranteed for 12 months. A pair of DIP shunts is also required to upgrade the CPU memory in the TRS-80 – these cost an additional $4.00. All kits come complete with full, step-by-step instructions which include labelled photographs. No soldering is required. You do not have to be an experienced electronic technician to instal them.

DISK DRIVE CABLES SUITABLE FOR ANY DISK DRIVES

DC-2 2 Drive Connector Cable . . . . . $39 incl. p&p
DC-4 4 Drive Connector Cable . . . . . $49 incl. p&p

DOUBLE THE SPEED AND CAPACITY OF YOUR DISK DRIVES PERCOM DOUBLER ONLY $220 plus $2.00 p&p

Installing a Doubler is like buying another set of disk drives, only much cheaper!! The doubler works with most modern disk drives including:- MPI, Micropolis, Pertec, TEAC (as supplied by Tandy). The doubler installs in the TRS-80 expansion interface, the System-80 expansion interface and the LNW Research expansion interface in a few minutes without any soldering, cutting of tracks, etc. It comes complete with its own TRSDOS compatible double density operating system.

DOUBLE-ZAP II – DOUBLE DENSITY PATCH FOR NEWDOS 80 ONLY $53.00 plus $1.00 p&p

DISCONTINUED

If you are using NEWDOS 80, then you also need DOUBLE-ZAP on diskette. This program upgrades your NEWDOS 80 to double density with ADR (automatic density recognition). It retains all the familiar features, including the ability to mix and match track counts on the same cable. In addition, it gives NEWDOS 80 the ability to mix densities on the same cable, automatically. If you place a single density diskette in drive Ø, say and a double density diskette in drive 1, Double-ZapII will recognise this and read/write to drive Ø in single density whilst at the same time it reads/writes to drive 1 in double density!

FLOPPY DOCTOR AND MEMORY DIAGNOSTIC (by MICRO CLINIC) $29.95 plus 50c. p&p

Two machine language programs on a diskette together with manual which thoroughly test your disk drives and memory. There are 19 possible error messages in the disk drive test and their likely causes are explained in the manual. Each pass of the memory tests checks every address in RAM 520 times, including the space normally occupied by the diagnostic program itself. When an error occurs the address, expected data, and actual data are printed out together with a detailed error analysis showing the failing bit or bits, the corresponding IC's and their location. This is the most thorough test routine available for TRS-80 disk users.

BOOKS

LEVEL II ROM REFERENCE MANUAL $24.95 + $1.20 p&p

Over 70 pages packed full of useful information and sample programs. Applies to both TRS-80 and SYSTEM 80.

TRS-80 DISK AND OTHER MYSTERIES $24.95 + $1.20 p&p

The hottest selling TRS-80 book in the U.S.A. Disk file structures revealed, DOS's compared and explained, how to recover lost files, how to rebuild crashed directories – this is a must for the serious Disk user and is a perfect companion to any of the NEWDOS's.

LEARNING LEVEL II $16.95 + $1.20 p&p

Written by Daniel Lien, the author of the TRS-80 Level I Handbook, this book teaches you, step-by-step, how to get the most from your Level II machine. Invaluable supplement to either the TRS-80 Level II Manual or the System-80 Manuals.

MORE AUSTRALIAN SOFTWARE

All programs designed to run on both the TRS-80 or the SYSTEM 80 without modification. Most programs include sound

TRIAD VOL 1 – L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Three separate games which test your powers of memory and concentration. The programs combine graphic displays and sound:

SIMON-SEZ: Just like the electronic music puzzles on sale for more than $20. Numbers are flashed on the screen and sounded in a sequence determined by the computer. Your task is to reproduce the sequence, correctly.

LINE?: Rather like a super, complicated version of noughts and crosses. You may play against another player or against the computer itself. But beware, the computer cheats!

SUPER CONCENTRATION: Just like the card game but with more options. You must find the hidden pairs. You may play against other people, play against the computer, play on your own, or even let the '80 play on its own.

TRIAD VOL 2 – L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Remember those "NUMERO" puzzles in which you had a matrix of numbers (or letters) with one blank space and you had to shuffle the numbers around one at a time until you had made a particular pattern? Well, **SHUFFLEBOARD**, the first program in this triad, is just this, except that the computer counts the number of moves you take to match the pattern it has generated – so it is not possible to cheat.

MIMIC is just like SHUFFLEBOARD except that you only see the computer's pattern for a brief span at the beginning of the game, then you must remember it!

In **MATCHEM**, you have to manoeuvre 20 pegs from the centre of the screen to their respective holes in the top or bottom rows. Your score is determined by the time taken to select a peg, the route taken from the centre of the screen to the hole and your ability to direct the peg into the hole without hitting any other peg or the boundary.

VISURAMA L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Two programs which give fascinating, ever-changing patterns on the screen.

LIFE is the fastest implementation of the Game of Life you will see on your '80. Machine language routines create up to 1200 new generations per minute for small patterns or up to 100 per minute for the full 128 x 48 screen matrix. Features full horizontal and vertical wraparound.

EPICYCLES will fascinate you for hours. The ever-changing ever-moving patterns give a 3D effect and were inspired by the ancient Greek theories of Ptolemy and his model of the Solar system.

EDUCATION AND FUN – L1/4K, L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Written by a primary school teacher to make learning enjoyable for his pupils, there are five programs in both Level I and Level II to suit all systems:

BUG-A-LUG: a mathematics game, in which you must get the sum correct before you can move.

AUSTRALIAN GEOGRAPHY: learn about Australian States and towns, etc.

SUBTRACTION GAME: build a tower with correct answers.

HOW GOOD IS YOUR MATHS? Select the function (+, –, ÷ or X) and degree of difficulty.

HANGMAN: That well known word game now on your computer.

Recommended for children from 6 to 9 years.

COSMIC FIGHTER & SPACE JUNK – L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Both programs have sound to complement their excellent graphics. In **COSMIC FIGHTER**, you must defend the earth against seven different types of alien aircraft. It is unlikely that you will be successful but you will have a lot of fun trying!

You mission in **SPACE JUNK** is to clean up all the debris left floating around in space by those other space games. It is not as simple as it sounds and space junk can be quite dangerous unless you are very careful.

SPACE DRIVE L2/4K & 16K
Cassette $8.95 Disk $13.95
+ 60c p&p

Try to manoeuvre your space ship through the meteor storms then land it carefully at the space port without running out of fuel or crashing. Complete with realistic graphics.

STARFIRE AND NOVA INVASION L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

Both programs include sound to improve their realism.

STARFIRE seats you in the cockpit of an X-wing fighter as you engage in battle with the deadly Darth Vader's Tie-fighters. Beware of the evil one himself and may the Force be with you.

In **NOVA INVASION**, you must protect your home planet of Hiberna from the invading NOVADIANS. You have two fixed guns at each side of the screen and a moveable one at the bottom. Apart from shooting down as many invaders as possible, you must protect your precious hoard of Vitaminium or perish!

AIR ATTACK AND NAG RACE – L2/16K
Cassette $10.95 Disk $15.95
+ 60c p&p

An unlikely combination of programs but they share the same author who has a keen sense of humour.

AIR ATTACK includes sound and realistic graphics. The aircraft even have rotating propellors! But they also drop bombs on you, so it's kill or be killed!

NAG RACE lets you pander to your gambling instinct without actually losing real money. Up to five punters can join in the fun. Each race results in a photo-finish whilst there is a visible race commentary at the bottom of the screen throughout the race. Happy punting!

FOUR LETTER MASTERMIND L2/16K
Cassette $8.95 Disk $13.95
+ 60c p&p

There are 550 four-letter words from which the computer can make its choice. You have 12 chances to enter the correct word. After each try, the computer informs you of the number of correct letters and those in the correct position. You can peek at the list of possible words but it will cost you points. Makes learning to spell fun.

MUSIC IV – L2/16K
Cassette $8.95 Disk $13.95
+ 60c p&p

Music IV is a music compiler for your '80. It allows you to compose or reproduce music with your computer that will surprise you with its range and quality. You have control over duration (full beat to 1/16 beat) with modifications to extend the duration by half or one third for triplets. Both sharps and flats are catered for as are rests. Notes on whole sections may be repeated. The program comes with sample data for a well-known tune to illustrate how it is done.

*****SAVE 00$'s ***SAVE 00$'s ***SAVE 00$'s ***MICRO-80 EXPANSION INTERFACE *****

MICRO-80's expansion interface utilises the proven LNW Research Expansion board. It is supplied fully built up and tested in an attractive cabinet with a self contained power supply, ready to plug in and go. The expansion interface carries MICRO-80's full, no hassle, 90-day warranty.

Features include:- • Sockets for up to 32K of memory expansion • Disk controller for up to 4 disk drives • Parallel printer port • Serial RS232C/20mA I/O port • Second cassette (optional)

The expansion interface connects directly to your TRS-80 L2/16K keyboard or, via SYSPAND-80 to your SYSTEM-80VIDEO GENIE

Prices: HD-010-A Expansion Interfaces with Ø K : $499.00 HD-010-B Expansion Interfaces with 32K : $549.00 HD-011 Data separator fitted (recommended) : add $29.00 HD-012 Dual cassette Interfaces fitted : add $19.00

The MICRO-80 Expansion Interface is also available in kit form.

Prices: HD-013 Kit consisting of LNW Research PC board and manual, ALL components including cabinet & power supply : $375.00 HD-011 Data separator for above $25.00 HD-013 Dual cassette Interface kit : $15.00

TURN THIS into this

for $49.00 plus $2.00 p & p

A choice of upper and lower case display is easier to read, gives greater versatility.

The Micro-80 lower case modification gives you this facility, plus the symbols for the 4 playing-card suits for $49.00 + $2.00 p. & p.

The Micro-80 modification features true below-the-line descenders and a block cursor.

Each kit comes with comprehensive fitting instructions and two universal lower-case drive routines on cassette to enable you to display lower case in BASIC programs.

The driver routines are self-relocating, self-protecting and will co-reside with other machine language programs such as Keyboard-debounce, serial interface driver programs etc.

Both programs give your TRS-80™ Model I or System 80™ an optional typewriter capability, i.e. shift for upper case.

The second programme also includes Keyboard-debounce and a flashing cursor.

You fit it. Or we can.

Fitting the modification requires soldering inside the computer. This should only be carried out by an experienced hobbyist or technician.

If you are at all dubious, a fitting service is available in all capital cities for only $20.00.

A list of installers is included with each kit.

Save $120 now.

ADD A DISK DRIVE TO YOUR TRS-80™ MODEL III FOR ONLY $875.00 OR ADD TWO FOR ONLY $1199.

The Micro-80 disk drive upgrade for the TRS-80™ Model III contains the following high quality components:

1 or 2 MPI 40-track single head disk drives, 1 VR Data double-density disk controller board and 1 dual drive power supply plus all the necessary mounting hardware, cables and comprehensive fitting instructions, which can be carried out with a minimum of fuss by any average computer owner.

Fitting service is available for $25.00 in most capital cities.

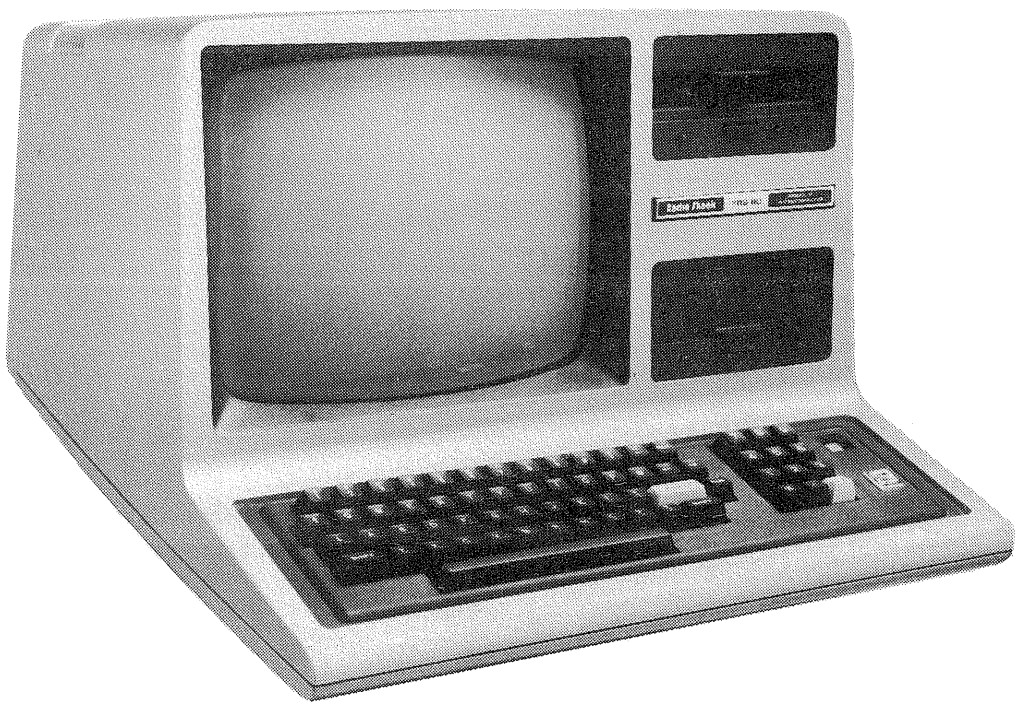
Daisy Wheel Typewriter/Printer

ONLY $2049 INC. S.T.

MICRO-80 has converted the new OLIVETTI ET-121 DAISY WHEEL typewriter to work with the TRS-80 and SYSTEM 80 or any other microcomputer with a Centronics parallel port (RS 232 serial interface available shortly). The ET-121 typewriter is renowned for its high quality, fast speed (17 c.p.s.), quietness and reliability. MICRO-80 is renowned for its knowledge of the TRS-80/SYSTEM 80 and its sensible pricing policy. Together, we have produced a dual-purpose machine:- an attractive, modern, correcting typewriter which doubles as a correspondence quality Daisy-wheel printer when used with your micro-computer.

How good is it? - This part of our advertisement was typeset using an ET-121 driven by a TRS-80. Write and ask for full details.

1.4 MEGABYTES ON LINE + 48K RAM
for $3800 incl. Sales Tax



MICRO-80's

MODEL 380 +

MICRO-80 has equipped the TRS-80 with two high reliability dual-head 80 track mini-floppy disk drives made by MPI, one of America's leading mini-disk drive manufacturers.

This turns the mild-mannered Model 3 into a powerhouse able to handle the most difficult business programs. The TRS-80 is one of the best-supported microcomputers in the world. MICRO-80 has been supporting the TRS-80 in Australia for 18 months and is one of Australia's leading dealers in MPI disk drives.

2.8 MEGABYTES FOR $5300 incl. Sales Tax

If you need even more file space you can add MICRO-80's external dual-drive cabinet enclosing two more dual-head 80 track drives for an additional $1500.

COMPUTER PRICES

MODEL 340
2 40 TRACK SINGLE HEAD DRIVES GIVING
350K FORMATTED STORAGE, 48K RAM — $2990 INCL. SALES TAX

MODEL 340 +
2 40 TRACK DUAL-HEAD DRIVES GIVING
700K FORMATTED STORAGE, 48K RAM — $3350 INCL. SALES TAX

MODEL 380
2 80 TRACK SINGLE HEAD DRIVES GIVING
700K FORMATTED STORAGE, 48K RAM — $3350 INCL. SALES TAX

MODEL 380 +
2 80 TRACK DUAL-HEAD DRIVES GIVING
1.4 MEGABYTE FORMATTED STORAGE, 48K RAM — $3800 INCL. SALES TAX

350K SYSTEM
MODEL 340, EPSON MX-80 PRINTER
NEWDOS 80 DISK OPERATING SYSTEM — $4070 INCL. SALES TAX

700K SYSTEM (40 Track)
MODEL 340 + , EPSON MX-80 PRINTER
NEWDOS 80 DISK OPERATING SYSTEM — $4429 INCL. SALES TAX

700K SYSTEM (80 Track)
MODEL 380, EPSON MX-80 PRINTER
NEWDOS 80 DISK OPERATING SYSTEM — $4429 INCL. SALES TAX

1.4 MEGABYTE SYSTEM
MODEL 380 + , EPSON MX-80 PRINTER
NEWDOS 80 OPERATING SYSTEM — $4880 INCL. SALES TAX

2.8 MEGABYTE SYSTEM
MODEL 380 + , DUAL EXTERNAL DRIVES,
MX-80 PRINTER, NEWDOS 80 OPERATING SYSTEM — $6380 INCL. SALES TAX

EXATRON STRINGY FLOPPY — $372.50 Incl. P&P

All Exatron Stringy Floppies sold by MICRO-80 include the special chained version of **HOUSEHOLD ACCOUNTS,** developed by Charlie Bartlett. When used on the ESF, this program is powerful enough to perform many of the accounting functions in a small business. Remember, the ESF comes complete with a comprehensive manual, a 2 way bus-extender cable, its own power supply and 10 wafers of mixed length. One wafer contains the Data Input/Output program and another the **HOUSEHOLD ACCOUNTS** program.

CAN'T MAKE UP YOUR MIND ABOUT THE ESF?

Then send in $5.00 for a copy of the manual. We will refund your $5.00 IN FULL when you purchase an ESF.

All prices subject to change without notice. Prices are F.O.B. Adelaide. All computers and systems carry MICRO-80's 90-day Warranty covering parts and labour.

SOFTWARE BY AUSTRALIAN AUTHORS

All our software is suitable for either the SYSTEM 80 or the TRS-80

NEW SOFTWARE FROM MICRO-80 PRODUCTS

BUSINESS PROGRAMS

MICROMANAGEMENT STOCK RECORDING SYSTEM (L2/16K)

Cassette version. . . . . $29.95 + $1.00 p&p
Stringy Floppy version . . . . . $33.95 + $1.00 p&p

This system has been in use for 9 months in a number of small retail businesses in Adelaide. It is therefore thoroughly debugged and has been tailor made to suit the requirements of a small business. MICROMANAGEMENT SRC enables you to monitor the current stock level and reorder levels of 500 different stock items per tape or wafer. It includes the following features:—

- Add new items to inventory
- Delete discontinued items from inventory
- List complete file
- Search for any stock number
- Save data to cassette or wafer
- Load data from cassette or wafer
- Adjusts stock levels from sales results and receipt of goods
- List all items requiring reordering

We can thoroughly recommend this program for the small business with a L2/16K computer.

SCOTCH BRAND COMPUTING CASSETTES

Super-quality personal computing cassettes.
C-10 pack of 10 ... ... $26.00 incl. p&p
C-30 pack of 10 ... ... $28.00 incl. p&p

UTILITIES

S-KEY by Edwin Paay $15.95 plus 50c. p&p

S-KEY is a complete keyboard driver routine for the TRS-80 and becomes part of the Level II basic interpreter. With S-KEY loaded the user will have many new features not available with the standard machine.

S-KEY features:

* S-KEY provides an auto-repeat for all the keys on the keyboard. If any key is held down longer than about half a second, the key will repeat until it is released.
* Graphic symbols can be typed direct from the keyboard, this includes all 64 graphic symbols available from the TRS-80/SYSTEM 80.
* S-KEY allows text, BASIC commands and/or graphics to be defined to shifted keys. This makes programming much easier as whole commands and statements can be recalled by typing shift and a letter key.
* Because S-KEY allows graphics to be typed directly from the keyboard, animation and fast graphics are easily implemented by typing the appropriate graphics symbols directly into PRINT statements.
* S-KEY allows the user to LIST a program with PRINT statements containing graphics, properly. S-KEY does this by intercepting the LIST routine when necessary.
* S-KEY allows the user to list an updated list of the shift key entries to the video display or line printer.
* S-KEY can be disabled and enabled when required. This allows other routines which take control of the keyboard to run with S-KEY as well.

Each cassette has TRS-80, DISK and SYSTEM 80 versions and comes with comprehensive documentation.

BMON by Edwin Paay $19.95 plus 50c. p&p

THE ULTIMATE HIGH MEMORY BASIC MONITOR L2/16-48K

Our own personnel refuse to write BASIC without first loading this amazing machine language utility program into high memory! BMON Renumbers; Displays BASIC programs on the screen while they are still loading; tells you the memory locations of the program just loaded; lets you stop a load part-way through; merges two programs, with automatic renumbering of the second so as to prevent any clashes of line numbers; recovers your program even though you did type NEW: makes one program invisible while you work on a second (saves hours of cassette time!); lists all the variables used in the program; makes SYSTEM tapes; lets you Edit memory directly . . . the list goes on and on. Cassette comes with 16K, 32K and 48K versions, ready to load. Can anyone afford NOT to have BMON?

EDUCATIONAL

RPN CALCULATOR (L2/16K & 32K)

$14.95 $ 50c. p&p

Give your computer the power of a $650 reverse polish notation calculator with 45 functions and selectable accuracy of 8 or 16 digits. The main stack and registers are continuously displayed whilst the menu is always instantly accessible without disturbing any calculations or register values. The cassette comes with both the 16K and 32K versions, the latter giving you the additional power of a programmable calculator. Comes with a very comprehensive 15 page manual, which includes instructions to load and modify the 32K programmable version to run in 16K. Whether for business or pleasure, this package will prove invaluable, and turn you '80 into a very powerful instrument.

GAMES

MICROPOLY (L2/16K) $8.95 + 60c p&p

Now you can play Monopoly on your micro. The old favourite board game has moved into the electronic era. This computer version displays the board on the screen, obeys all the rules and, best of all, the banker does not make mistakes with your change!

CONCENTRATION (L2/16K) $8.95 + 60c p&p

Another application of supergraphics. There are 28 "cards" displayed on the screen, face down. Players take it in turn to turn them over with the object of finding matching pairs. There are 40 different patterns which are chosen at random, so the game is full of endless variety. This is of particular value in helping young children to learn the art of concentrating and, at the same time, to introduce them to the computer.

METEOR AND TORPEDO ALLEY (L2/16K)

$10.95 + 60c p&p

Those who frequent games arcades will recognize these two electronic games. In METEOR you must destroy the enemy space ships before they see you. In its most difficult mode, the odds are a thumping 238 to 1 against you being successful. In torpedo alley you must sink the enemy ships without hitting your own supply ship. Both games include sound effects and are remarkably accurate reproductions of the arcade games.

AUSTRALIAN SOFTWARE (Cont.)

GAMES

SHEEPDOG (L2/16K) **$8.95 + 60c p&p**
Ever wondered how a sheepdog manages to drive all those awkward sheep into a pen? Well, here is your chance to find out just how difficult it is and have a lot of fun at the same time. You control the sheepdog, the computer controls the sheep! As if that isn't enough, look out for the dingoes lurking in the bush!

U BOAT **$8.95 + 60c p&p**
Real time simulation at its best! Comes with working sonar-screen and periscope, a full rack of torpedoes, plenty of targets, working fuel and battery meters, helpful Mothership for high-seas reprovisioning and even has emergency radio for that terrible moment when the depth charges put your crew at risk. Requires Level II/16K.

SPACE INVADERS WITH SOUND **$8.95 + 60c p&p**
Much improved version of this arcade favourite with redesigned laser and cannon blasts, high-speed cannon, 50 roving drone targets, 10 motherships and heaps of fun for all. Level II with 4K and 16K versions on this cassette.

GOLF (L2/16K) **$8.95 + 60c p&p**
Pit your skills of mini-golf against the computer. Choose the level of difficulty, the number of holes and whether you want to play straight mini golf or crazy golf. Complete with hazards, water traps, bunkers and trees. Great fun for kids of all ages.

DOMINOES(L2/16K) **$8.95 + 60c p&p**
Pit your skill at dominoes against the computer, which provides a tireless opponent. Another application of supergraphics from the stable of Charlie Bartlett. Dominoes are shown approximately life size in full detail (except for colour!). The monitor screen is a window which you can move from one end of the string of dominoes to the other. Best of all, you don't lose any pieces between games!

KID'S STUFF (formerly MMM-1) **$8.95 + 60c p&p**
Three games on one cassette from that master of TRS-80 graphics, Charlie Bartlett. Includes INDY 500, an exciting road race that gets faster and faster the longer you play, SUBHUNT in which your warship blows up unfortunate little submarines all over the place, and KNIEVEL (as in motorcycle, ramp and buses).

OTHER PROGRAMS

INFINITE BASIC BY RACET (32K/1 DISK)
$49.95 + 50c. p&p
Full matrix functions – 30 BASIC commands; 50 more STRING functions as BASIC commands.

GSF/L2/48K **$24.95 + 50c. p&p**
18 machine language routines including RACET sorts.

BUSINESS ADDRESS AND INFORMATION SYSTEM (48K/DISK) **$24.95 + 50c. p&p**
Allows you to store addresses and information about businesses, edit them and print them out.

HISPED (L216, 32 or 48K) $29.95
This machine language program allows you to SAVE and LOAD programs and data to tape at speeds up to 2000 band (4 times normal) using a standard cassette recorder. A switch must be installed to remove the XRX III loading board, if fitted.

LOWER CASE FOR YOUR TRS-80/SYSTEM 80
Kit only $49.00 plus $2.00 p&p

Give your TRS-80 or SYSTEM 80 a lower case display with proper descenders and a block cursor (similar to the TRS-80 Model III). Also includes symbols for the four suits of cards. Includes full fitting instructions, all necessary components and a special machine language driver program to enable lower case in BASIC. The modification is similar to the Tandy model and does not work with Electric Pencil without further modifications.

These kits require disassembly of your computer and some soldering. They should only be installed by someone who has experience in soldering integrated circuits, using a low power, properly earthed soldering iron. If you do not have the necessary experience/equipment, we will install the modification for you for $20 plus freight in both directions. Make sure you arrange the installation with us first, before despatching your computer, so that we can assure you of a rapid turn-around. We are also arranging to have installers in each State. See elsewhere in this issue for their names and addresses.

PRICES

Cat No.
HD-020 Lower case mod kit for TRS-80
$49.00 plus $2.00 p&p
HD-021 Lower case mod kit for SYSTEM-80
$49.00 plus $2.00 p&p

EPSON MX-80 PRINTER
ONLY *$949 Inc. Cable for TRS-80 and p&p
(*Printer only – $940 incl. p&p)

The EPSON MX-80 printer is compact, quiet, has features unheard of only 2-3 years ago in a printer at any price and, above all, is ultra-reliable. All available print modes may be selected under software control. Features include:

- high quality 9x9 dot-matrix character formation
- 3 character densities
  - . 80 characters per line at 10 chars/inch
  - .132 characters per line at 16.5 chars/inch
  - . 40 characters per line at 5 chars/inch
- 2 line spacings
  - . 6 lines per inch 8 lines per inch
- 80 characters per second print speed
- bi-directional printing
- logical seeking of shortest path for printing
- lower case with descenders
- TRS-80 graphics characters built in
- standard Centronics printer port

The bi-directional printing coupled with the logical seeking of the shortest print path (which means that the print head will commence printing the next line from the end which requires the least travel, thereby minimising unutilised time) gives this printer a much higher throughput rate than many other printers quoting print speeds of 120 c.p.s. or even higher.

GREEN SCREEN SIMULATOR
$9.50 incl. p&p

The GREEN SCREEN SIMULATOR is made from a deep green perspex, cut to fit your monitor. It improves contrast and is much more restful to the eyes than the normal grey and white image.

All editorial staff of MICRO-80 are now using GREEN SCREEN SIMULATORS on their own monitors.

Please make sure to specify whether you have an old (squarish) or new (rounded) style monitor when ordering. Not available for Dick Smith monitors.

```
                00240 ;SEEK TOKENS IN LINE
4056 D7         00250 NEXCHR  RST     10H             ;A=NEXT BYTE, SET FLAGS
4057 FE00       00260 KEEPON  CP      0               ;EOL REACHED?
4059 281E       00270         JR      Z,NEXT          ;TO NEXT LINE IF EOL
405B EB         00280         EX      DE,HL           ;SAVE HL
405C 217C40     00290         LD      HL,TOKENS       ;HL=TOKENS ORG
405F 010700     00300         LD      BC,7            ;BC=NO. OF TOKENS
4062 EDB1       00310         CPIR                    ;SET Z IF A IS TOKEN
4064 EB         00320         EX      DE,HL           ;RESTORE HL
4065 20EF       00330         JR      NZ,NEXCHR       ;KEEP SEEKING TOKENS
                00340 ;SEEK LINE REF AFTER TOKEN
4067 D7         00350 FOUND   RST     10H             ;A=NEXBYT,SET C IF NUMRIC
4068 30ED       00360         JR      NC,KEEPON       ;NO MORE LINE REFS
                00370 ;VALIDATE LINE REFS
406A CD5A1E     00380         CALL    1E5AH           ;DE=LINE NO.,(HL)=DELIMTR
406D E5         00390         PUSH    HL              ;SAVE HL
406E CD2C1B     00400         CALL    1B2CH           ;SET C IF LINE EXISTS
4071 E1         00410         POP     HL              ;RESTORE HL
4072 D24A1E     00420         JP      NC,1E4AH        ;FC ERROR ROUTINE
4075 AF         00430         XOR     A               ;A=0
4076 BE         00440         CP      (HL)            ;IS DELIMITER EOL?
4077 20EE       00450         JR      NZ,FOUND        ;SEEK FURTHER REFS
4079 23         00460 NEXT    INC     HL              ;SKIP EOL
407A 18CE       00470         JR      NEXLIN          ;SEARCH NEXT LINE
                00480 ;DATA - COMPRESSION TOKENS
407C 918D       00490 TOKENS  DEFW    8D91H           ;GOTO, GOSUB
407E 95CA       00500         DEFW    0CA95H          ;THEN, ELSE
4080 8E9F       00510         DEFW    9F8EH           ;RESUME, RUN
4082 B6         00520         DEFB    0B6H            ;DELETE
4047            00530         END     VALID
                00540 ;THE ABOVE ROUTINE IS RELOCATABLE WITHOUT RE-ASSEMBLY.
                00550 ;N.B. IT MUST NOT BE USED WITH PROGRAMS THAT CONTAIN
                00560 ;GRAPHICS BYTES IN STRING CONSTANTS OR REMARKS.
                00570 ;IT OCCUPIES 59 BYTES
```

***** MULTIPLE REGRESSION ANALYSIS L2/16K by T.R. Jones *****

Multiple regression analysis is used to test the relationship between a dependent variable (Y) and a number of independent variables(X1,X2,X3,....XN).

This routine allows multiple regression analysis between the independent variable Y and two dependent variables X1 and X2, in accordance with the model:-

$$Y = A + BX1 + CX2.$$

The routine calculates the coefficients for the multiple regression equation, the mean values of X1, X2 and Y, the coefficient of multiple regression (an indication of closeness of fit, where 1 is perfect), and the percentage variation in Y due to X1, X2, and X1 and X2 jointly. In addition, a residual table can be printed on command and estimations of Y computed for entered values of X1 and X2.

The routine will accept up to 100 data sets, which are verified and can be edited following entry.

```
20 CLS:DEFINTI-N:DIMX(3,100),A(6,3),R(6,3):P$="######.####"
30 PRINTTAB(18)"MULTIPLE REGRESSION ANALYSIS":
   PRINTTAB(18)STRING$(28,45)
40 PRINT:PRINT:PRINT:PRINTTAB(29)"MODEL":PRINT:
   PRINTTAB(23)"Y = A + BX1 + CX2"
50 PRINT@977,"TYPE <ENTER> TO CONTINUE ....";:INPUTI$
60 CLS:PRINTTAB(27)"ENTER DATA":PRINT:PRINT:
   PRINTTAB(13)"NUMBER OF DATA POINTS - MAXIMUM IS 100"
70 PRINTTAB(28)"*";:INPUTND:IFND<2ORND>100
   PRINTCHR$(27);CHR$(30);CHR$(29);CHR$(27):GOTO70
80 PRINT:PRINT:FORI=1TOND:
   PRINT"X1(";I;") =";:INPUTX(1,I):
   PRINTTAB(22)CHR$(27)"X2(";I;") =";:INPUTX(2,I):
   PRINTTAB(44)CHR$(27)"Y(";I;") =";:INPUTX(3,I):NEXTI
90 CLS:K=0:PRINTTAB(26)"DATA ENTERED":PRINT:FORI=1TOND:K=K+1:
   PRINT"X1(";I;") =";X(1,I);TAB(22)"X2(";I;") =";X(2,I);
   TAB(44)"Y(";I;") =";X(3,I):IFK<10NEXTI
   ELSEPRINT@977,"TYPE <ENTER> TO CONTINUE ...";:
   INPUTI$:K=0:CLS:NEXTI
100 PRINT@983,"EDIT DATA (Y/N)";
```

```
110 I$=INKEY$:IFI$=""THEN110ELSEIFI$="Y"THEN120ELSEIFI$="N"
   THEN140ELSE110
120 CLS:INPUT"DATA POINT TO BE EDITED";I:
   IFI>NDTHEN120ELSEPRINTTAB(4)"X1(";I;") =";X(1,I);
   TAB(24)"X2(";I;") =";X(2,I);
   TAB(44)"Y(";I;") =";X(3,I)
130 PRINT"NEW X1(";I;") =";:INPUTX(1,I):
   PRINTTAB(24)CHR$(27);"X2(";I;") =";:INPUTX(2,I):
   PRINTTAB(44)CHR$(27);"Y(";I;") =";:INPUTX(3,I):GOTO90
140 CLS:PRINT@473,"COMPUTING ...."
150 FORI=1TO6:FORJ=1TO3:A(I,J)=0:R(I,J)=0:NEXTJ,I
160 FORI=1TOND:A(1,1)=A(1,1)+X(1,I):A(1,2)=A(1,2)+X(2,I):
   A(1,3)=A(1,3)+X(3,I):NEXTI:A(2,1)=A(1,1)/ND:
   A(2,2)=A(1,2)/ND:A(2,3)=A(1,3)/ND
170 FORI=1TOND:A(3,1)=A(3,1)+(X(1,I)[2):A(3,2)=A(3,2)+(X(2,I)[2)
:   A(3,3)=A(3,3)+(X(3,I)[2):NEXTI
180 FORI=1TOND:A(4,1)=A(4,1)+(X(1,I)*X(3,I)):
   A(4,2)=A(4,2)+(X(2,I)*X(3,I)):A(4,3)=A(4,3)+(X(1,I)*X(2,I)):
   NEXTI
190 A(5,1)=A(3,1)-(A(1,1)[2/ND):A(5,2)=A(3,2)-(A(1,2)[2/ND):
   A(5,3)=A(3,3)-(A(1,3)[2/ND)
200 A(6,1)=A(4,1)-((A(1,1)*A(1,3))/ND):
   A(6,2)=A(4,2)-((A(1,2)*A(1,3))/ND):
   A(6,3)=A(4,3)-((A(1,1)*A(1,2))/ND)
210 R(1,1)=(A(6,1)[2)/(A(5,3)*A(5,1)):R(1,2)=SQR(R(1,1)):
   IFA(6,1)<OTHENR(1,2)=-R(1,2)
220 R(2,1)=(A(6,2)[2)/(A(5,3)*A(5,2)):R(2,2)=SQR(R(2,1)):
   IFA(6,2)<OTHENR(2,2)=-R(2,2)
230 R(3,1)=(A(6,3)[2)/(A(5,1)*A(5,2)):R(3,2)=SQR(R(3,1)):
   IFA(6,3)<OTHENR(3,2)=-R(3,2)
240 R(4,1)=(R(1,2)-(R(2,2)*R(3,2)))[2:
   R(4,1)=R(4,1)/((1-R(2,1))*(1-R(3,1))):
   IFR(4,1)<OTHENR(4,2)=0ELSER(4,2)=SQR(R(4,1))
250 R(5,1)=(R(2,1)-(R(2,2)*R(3,2)))[2:
   R(5,1)=R(5,1)/((1-R(1,1))*(1-R(3,1))):
   IFR(5,1)<OTHENR(5,2)=0ELSER(5,2)=SQR(R(5,1))
260 RS=R(2,1)+R(1,1)-(2*R(1,2)*R(2,2)*R(3,2)):
   RS=RS/(1-R(3,1)):IFRS>1THENRS=1:RM=1ELSERM=SQR(RS)
270 C=(A(6,3)*A(6,1))-(A(5,1)*A(6,2)):
   C=C/(A(6,3)[2-(A(5,2)*A(5,1)))
280 B=A(6,1)-(C*A(6,3)):B=B/A(5,1)
290 D=A(2,3)-(B*A(2,1))-(C*A(2,2))
300 CLS:PRINT"THE EQUATION FOR THE REGRESSION LINE IS :":PRINT:
   PRINT"YHAT = ";
310 IFD<0PRINT"-";ABS(D);ELSEPRINTD;
320 IFB<0PRINT"-";ABS(B);"X1";ELSEPRINT"+";B;"X1";
330 IFC<0PRINT" -";ABS(C);"X2";ELSEPRINT" +";C;"X2";
340 PRINT:PRINT:PRINT"MEAN X1 =";A(2,1):
   PRINT"MEAN X2 =";A(2,2):
   PRINT"MEAN Y  =";A(2,3)
350 PRINT:PRINT"COEFFICIENT OF MULTIPLE CORRELATION =";RM
360 R1=R(1,1)*100:R2=R(2,1)*100:RS=RS*100:PRINT:
   PRINT"VARIATION IN Y DUE TO : X1 =";R1;"%":
   PRINTTAB(24)"X2 =";R2;"%":
   PRINTTAB(19)"X1 & X2 =";RS;"%"
370 PRINT@978,"PRINT RESIDUAL TABLE (Y/N)";
380 I$=INKEY$:IFI$=""THEN380ELSEIFI$="Y"THEN390ELSEIFI$="N"
   THEN470ELSE380
390 CLS:J=0
400 FORI=1TOND:YH=D+(B*X(1,I))+(C*X(2,I)):RE=YH-X(3,I)
410 IFJ=0GOSUB510
420 PRINTUSING"##";I;:PRINTTAB(5);:PRINTUSINGP$;X(1,I);X(2,I);
   X(3,I);YH;RE
430 J=J+1:IFJ>9THEN440ELSE450
440 PRINT@977,"TYPE <ENTER> TO CONTINUE ...";:INPUTI$:J=0:CLS
450 NEXTI
460 RS=RS/100:SD=(1-RS)*A(5,3):PRINT:
   PRINT"SUM OF SQUARES OF DEVIATION FROM REGRESSION =";SD
470 PRINT@978,"ESTIMATE VALUES OF Y (Y/N)";
480 I$=INKEY$:IFI$=""THEN480ELSEIFI$="Y"THEN490ELSEIFI$="N"
   THEN460ELSE480
490 CLS
500 INPUT"X1";XA:PRINTTAB(21)CHR$(27);:INPUT"X2";XB:
   YH=D+(B*XA)+(C*XB):PRINTTAB(42)CHR$(27);"Y =";YH:GOTO500
510 CLS:PRINT"POINT";TAB(11)"X1";TAB(22)"X2";TAB(33)"Y";
    TAB(43)"YHAT";TAB(55)"RES":PRINT:RETURN
```

***** ATOMIC TABLES - L2/16K by D. Balaic *****

As I read John S. Bone's article from the U.K., I realised how poor the Australian government's reaction to computers has been. Where America is already using computers in education wholesale from second grade upwards, only a few technical colleges have PETs or the such here. When I saw the possibilities of computers in teaching mathematics and sciences, I felt I simply had to write an educational program to demonstrate. So "Atomic Tables" was evolved. This program not only aims to teach, but also can act as a reference to students studying chemistry, for it shows not only the name, symbol and element atomic number, but also gives the group or series and accurately gives the electon shells of the said atom. As the movements of atoms in the different shells (or energy levels) is not always easily predictable, this program is of particular usefulness. It is also capable of telling either of name, number or symbol from any of these given. The program uses one large data array to find all three particulars of an atom. Once it finds them it prints those which are required by the user. When it completes one question display, it searches its data for the required element, during which time the user has to try to think of the element. If the user wants only to find the data, there will be a short wait before the answer is displayed.

```
20 DIMD$(312)
30 'ELEMENT TABLES
40 CLS:PRINT@24,"THE ELEMENTS";:PRINT@89,"MAIN INDEX";:PRINT@192
,"YOU MAY CHOOSE ANY OF THE FOLLOWING:";:PRINT@330,"A   -TO OBTAI
N AN ELEMENT'S ATOMIC NO.";:PRINT@394,"B   -TO OBTAIN AN ELEMENT'
S SYMBOL";:PRINT@458,"C   -TO OBTAIN AN ATOMIC NO.'S ELEMENT";
50 PRINT@522,"D   -TO OBTAIN AN ATOMIC NO.'S SYMBOL";:PRINT@586,
"E   -TO OBTAIN A SYMBOL'S ATOMIC NO.";:PRINT@650,"F   -TO OBTAIN
 A SYMBOL'S ELEMENT";:PRINT@714,"G   -FOR A LIST OF ALL RELEVANT
DATA ASSOCIATED WITH                AN ELEMENT.";
60 A$=INKEY$:IFA$=""THEN60ELSEIFA$="A"THEN140ELSEIFA$="B"THEN90E
LSEIFA$="C"THEN70ELSEIFA$="D"THEN190ELSEIFA$="E"THEN210ELSEIFA$="
F"THEN260ELSEIFA$="G"THEN310ELSE60
70 CLS:PRINT@463,"PLEASE ENTER ATOMIC NO.";:INPUTA:GOSUB600:FORN
=1TOA*3+2:READD$:NEXT:CLS:PRINT@464,"THE ELEMENT FOR"A"IS "D$;:PR
INT@979,"HIT ANY KEY TO CONTINUE";:RESTORE
80 IFINKEY$=""THEN80ELSE30
90 CLS:PRINT@466,"PLEASE ENTER ELEMENT";:INPUTF$:GOSUB600:FORN=1
TO312:READD$(N):IFD$(N)=F$THEN120ELSENEXT
100 GOSUB160
110 GOTO90
120 CLS:N=N+1:READD$(N):PRINT@465,"THE SYMBOL OF "F$" IS "D$(N):
PRINT@979,"HIT ANY KEY TO CONTINUE";:RESTORE
130 IFINKEY$=""THEN130ELSE30
140 CLS:PRINT@465,"PLEASE ENTER ELEMENT";:INPUTS$:GOSUB600:FORN=
1TO312:READD$(N):IFD$(N)=S$THEN170ELSENEXT:GOSUB160
150 GOTO140
160 CLS:RESTORE:PRINT@470,"SORRY, TRY AGAIN";:FORJ=1TO870:NEXT:R
ETURN
170 CLS:PRINT@464,"THE ATOMIC NO. OF "S$" IS "D$(N-1);:PRINT@979
,"HIT ANY KEY TO CONTINUE";:RESTORE
180 IFINKEY$=""THEN180ELSE30
190 CLS:PRINT@466,"PLEASE ENTER ATOMIC NO.";:INPUTA:GOSUB600:FOR
N=1TOA*3+3:READD$:NEXT:CLS:PRINT@469,"THE SYMBOL OF"A"IS "D$;:PRI
NT@979,"HIT ANY KEY TO CONTINUE";:RESTORE
200 IFINKEY$=""THEN200ELSE30
210 CLS:PRINT@466,"PLEASE ENTER SYMBOL";:INPUTS$:GOSUB600:FORN=1
TO312:READD$(N):IFD$(N)=S$THEN240ELSENEXT
220 GOSUB160
230 GOTO210
240 CLS:PRINT@464,"THE ATOMIC NO. OF "S$" IS "D$(N-2);:PRINT@979
,"HIT ANY KEY TO CONTINUE";:RESTORE
250 IFINKEY$=""THEN250ELSE30
260 CLS:PRINT@466,"PLEASE ENTER SYMBOL";:INPUTS$:GOSUB600:FORN=1
TO312:READD$(N):IFD$(N)=S$THEN290ELSENEXT
270 GOSUB160
280 GOTO260
290 CLS:PRINT@463,"THE ELEMENT FOR "S$" IS "D$(N-1);:PRINT@979,"
HIT ANY KEY TO CONTINUE";:RESTORE
300 IFINKEY$=""THEN300ELSE30
310 CLS:PRINT@452,"PLEASE ENTER EITHER ATOMIC NO., SYMBOL OR ELE
MENT";:INPUTX$:GOSUB600:IFVAL(X$)<>ON=VAL(X$):IFN>103THEN590ELSE3
40ELSEJ=LEN(X$)
320 FORN=1TO312:READD$(N):IFD$(N)=X$THEN330ELSENEXT:RESTORE:GOTO
310
330 RESTORE:N=N/3:N=INT(N):IFJ<=2N=N-1
340 FORNN=1TON*3+1:READD$(NN):NEXT:CLS:PRINT@7,"ATOMIC NO.:"N",
ELEMENT: ";:READD$(NN):PRINTD$(NN)", SYMBOL: ";:NN=NN+1:READD$(NN
):PRINTD$(NN)".":PRINT@72,"NO. OF PROTONS:"N", ELECTRONS: "N"."
```

```
350 PRINT@670,N"P";:GOSUB390:PRINT@320,"GROUP:";:PRINT@384,GN$;:
PRINT@604,".-K="K"-.";:PRINT@540,".-L="L"-.";:PRINT@476,".-M="M"-
.";:PRINT@412,".-N="P"-.";:PRINT@348,".-O="Q"-.";:PRINT@284,".-P=
"R"-.";:PRINT@220,".-Q= 0 -.";:RESTORE
360 PRINT@979,"HIT ANY KEY TO CONTINUE";:K=0:L=0:M=0:R=0:P=0:Q=0
370 IFINKEY$=""THEN370ELSE30
380 DATA541,544,
390 GN$="TRANSITIONAL ELEMENTS":IFN=30RN=110RN=190RN=370RN=550RN
=87GN$="I"ELSEIFN=40RN=120RN=200RN=380RN=560RN=88GN$="II"ELSEIFN=
50RN=130RN=310RN=490RN=81GN$="III"ELSEIFN=60RN=140RN=320RN=500RN=
82GN$="IV"ELSEIFN=70RN=150RN=330RN=510RN=83GN$="V"
400 IFN=80RN=160RN=340RN=520RN=84GN$="VI"ELSEIFN=90RN=170RN=350R
N=530RN=85GN$="VII"ELSEIFN=20RN=100RN=180RN=360RN=540RN=540RN=866
N$="VIII OR 0 - NOBLE GASES"ELSEIFN=1GN$="HYDROGEN":K=0:R=0:Q=0
410 IFN>=90GN$="ACTINIDE SERIES"ELSEIFN>=58ANDN<=71GN$="LANTHANI
DE SERIES"
420 IFN>=2K=2:N=N-2ELSEK=N:GOTO480
430 IFN>=8L=8:N=N-8ELSEL=N:GOTO480
440 IFN>=18M=18:N=N-18ELSEM=N:GOTO480
450 IFN>=32P=32:N=N-32ELSEP=N:GOTO480
460 IFN>=32Q=32:N=N-32ELSEQ=N:GOTO480
470 IFN>=18R=18:N=N-18ELSER=N
480 N=K+L+M+P+Q+R:IFN=19M=M-1:P=1:GOTO580
490 IFN>=20ANDN<=30M=M-2:P=2:GOTO580
500 IFN=37P=P-1:Q=1:GOTO580
510 IFN=61P=P-1:Q=2:GOTO580
520 IFN=62Q=2:GOTO580
530 IFN>=38ANDN<=62P=P-2:Q=2:GOTO580
540 IFN=69Q=Q-1:R=1:GOTO580
550 IFN=93Q=Q-1:R=2:GOTO580
560 IFN=94R=2:GOTO580
570 IFN>=70ANDN<=94Q=Q-2:R=2:GOTO580
580 RETURN
590 REM
600 CLS:PRINT@472,"COMPILING DATA";:RETURN
610 DATA1,HYDROGEN,H,2,HELIUM,HE,3,LITHIUM,LI,4,BERYLIUM,BE,5,BO
RON,B,6,CARBON,C,7,NITROGEN,N,8,OXYGEN,O,9,FLOURINE,F,10,NEON,NE,
11,SODIUM,NA,12,MAGNESIUM,MG,13,ALUMINIUM,AL,14,SILICON,SI,15,PHO
SPHORUS,P,16,SULPHUR,S,17,CHLORINE,CL,18,ARGON,A,19,POTASSIUM,K
620 DATA20,CALCIUM,CA,21,SCANDIUM,SC,22,TITANIUM,TI,23,VANADIUM,
V,24,CHROMIUM,CR,25,MANGANESE,MN,26,IRON,FE,27,COBALT,CO,28,NICKE
L,NI,29,COPPER,CU,30,ZINC,ZN,31,GALLIUM,GA,32,GERMANIUM,GE,33,ARS
ENIC,AS,34,SCLENIUM,SE,35,BROMINE,BR,36,KRYPTON,KR,37,RUBIDIUM,RB
630 DATA38,STRONTIUM,SR,39,YTTRIUM,Y,40,ZIRCONIUM,ZR,41,NIOBIUM,
NB,42,MOLYBDENUM,MO,43,TECHNETIUM,TC,44,RUTHENIUM,RU,45,RHODIUM,R
H,46,PALLADIUM,PD,47,SILVER,AG,48,INDIUM,IN,49,CADMIUM,CD,50,TIN,
SN,51,ANTIMONY,SB,52,TELLURIUM,TE
640 DATA53,IODINE,I,54,XENON,XE,55,CAESIUM,CS,56,BARIUM,BA,57,LA
NTHANUM,LA,58,CERIUM,CE,59,PRASEODYMIUM,PR,60,NEODYMIUM,ND,61,PRO
METHIUM,PM,62,SAMARIUM,SM,63,EUROPIUM,EU,64,GADOLINIUM,GD,65,TERB
IUM,TB,66,DYSPROSIUM,DY,67,HOLMIUM,HO
650 DATA68,ERBIUM,ER,69,THULIUM,TM,70,YTTERBIUM,YB,71,LUTETIUM,L
U,72,HAFNIUM,HF,73,TANTALUM,TA,74,TUNGSTEN,W,75,RHENIUM,RE,76,OSM
IUM,OS,77,IRIDIUM,IR,78,PLATINUM,PT,79,GOLD,AU,80,MERCURY,HG,81,T
HALLIUM,TL,82,LEAD,PB,83,BISMUTH,BI
660 DATA84,POLONIUM,PO,85,ASTATINE,AT,86,RADON,RN,87,FRANCIUM,FR
,88,RADIUM,RA,89,ACTINIUM,AC,90,THORIUM,TH,91,PROTOACTINIUM,PA,92
,URANIUM,U,93,NEPTUNIUM,NP,94,PLUTONIUM,PU,95,AMERICIUM,AM,96,CUR
IUM,CM,97,BERKELIUM,BK,98,CALIFORNIUM,CF
670 DATA99,EINSTEINIUM,ES,100,FERNIUM,FM,101,MENDELEVIUM,MV,102,
NOBELIUM,NO,103,LAWRENCIUM,LW
```

***** TEXT TYPER L2/16K by M.J. Leonard *****

FEATURES -

* Standard keyboard gives lower-case with visual indication of shifted upper-case.
* Line justification or non-justification with non-justifying space provision.
* Indentation setting, readily changeable.
* Line change, line delete.
* Word change, word delete.
* Lines insert, word insert.

* Centering of text.
* Review text.
* Line counter.
* Record to tape.
* Load from tape.
* 'Packed' data displayed on reading and recording tape.
* Print draft document with line numbers.
* Print finished text.
* Prints on A4 sheet or roll paper.

This program was developed on a System 80. TRS-80 users should use the arrow keys as shown below:

'ESC' = ↑
'CTRL' = ↓
'TAB' = →
'BACKSPACE' = ←

This program processes text line by line with a maximum input at one time of 9Ø lines.

Each line consists of 62 characters - indicated by a staff on the screen. Upon RUNning the program, selection of an indentation position is made and indicated on staff. After selection of 'justification' or 'non-justification' entry can proceed.

If right-hand justification is selected, any line of characters finishing within 7 spaces of the end of the staff, will be justified. Upon ENTERing the line a string of "J"s will indicate the extra spaces being inserted to bring about justification. If the relationship between two or more words is to be maintained then CTRL must be used instead of the SPACE-BAR when inserting a space between these words. (Even with a line of 62 characters there are 3 spaces of justification - this is important for later word correction).

If change of indentation is required (for sub-paragraphing or tabulation) call up MENU and enter 'Indent' for 'Choice'. This also allows change to and from line justification.

To centre text for headlines etc., type material at beginning of staff and press ESC instead of ENTER. This will centre text and advance to next line input.

To call up MENU 'Shift' and 'ESC' keys are depressed. This then allows selection of the following:-

1. PRINT DRAFT DOCUMENT

   Line 192Ø in the program has been written for an Okidata 'Microline 8Ø' printer and selects 16.5 characters/inch, 1Ø5 characters to a line and 4 lines per inch. This spacing plus the printing of the line numbers allows easy corrections and additions to be made. Forty lines are printed on A4 paper before printing stops and a prompt is given to insert more paper.

2. PRINT COMPLETED TEXT

   Again line 2Ø1Ø sets the 'Microline 8Ø' for 1Ø characters/inch, 64 characters to a line and 6 lines per inch. This gives clean clear final copy. When printing in this style prompt for new paper comes after 59 lines. (Text will be lost if all pages are not printed - for other printers, both these program lines will need changing).

3. RECORD TO TAPE
4. LOAD FROM TAPE

   Upon selecting these facilities, data is packed into groups of three lines, commas and colons changed to graphic characters so as not to change formatting, and recording and loading shown on the screen. (Two dumps are recommended to allow for tape faults).

5. REVIEW DATA

   Line by line review is carried out by pressing ENTER for each line: pressing any other key will return the MENU.

6. CHANGE DATA

   A request for the line to be altered is given. Upon entry of this number the selection of whole line or individual word change is made. Selecting Word Change gives the selected

line again, which is disassembled and repeated. The following keys allow corrections and insertions. (If any change causes the line to exceed 64 characters, the line is printed ending in a '+' sign, indicating lost words. These words must be added on an inserted line):

TAB - presents each word for inspection.
BACK SPACE - removes the previous word from screen.
CTRL - deletes the last word displayed.
ESC - presents a small block character and allows insertion of a new word. During word insertion 'Shift-@' will insert a space.
SPACE BAR - fixes the new word into the line and allows further word inspection.
'@' - allows escape from the line being changed, rejustification taking place if necessary.

7. ADD TO DOCUMENT

This presents the last three lines of the previously entered text and allows further line entry.

8. INSERT NEW LINE

The prompt asks for the line number to be inserted. This will be the line number after the desired insert. If correct placement is shown, type in new line and program returns to MENU.

9. DELETE A LINE

The selected line is displayed and is deleted by pressing ENTER. Line numbering is moved up one for all lines higher than the deletion, therefore when more than one deletion is required, the highest number should be removed first.

The program is error-trapped to return to MENU. If the program is lost, however, GOTO 17ØØ will usually allow recovery.

```
100 CLS:PRINTTAB(25)CHR$(34);"TEXT-TYPER";CHR$(34):PRINTTAB(25)S
TRING$(12,42):PRINTTAB(37);"WRITTEN BY M.J. LEONARD.":PRINT
110 PRINT:PRINT"SELECT INDENTATION AND LINE JUSTIFICATION":PRINT
:PRINT"ENTER:  I   TO CONTINUE":PRINT
120 Z$=INKEY$:IF Z$="" THEN GOTO 120
130 CLEAR 5000:DIM A$(90),T$(20),B$(90)
140 ON ERROR GOTO 1470
150 REM INDENTATION POSITION
160 TB%=0
170 PRINT"ENTER INDENTATION";TAB(25);
180 X$=INKEY$:IF X$="" THEN GOTO 180
190 X%=ASC(X$)
200 IF X%=13 THEN PRINT:GOTO 250
210 X1$=X1$+X$:PRINT X$;
220 IF VAL(X1$)<0 OR VAL(X1$)>62 THEN PRINT CHR$(29);CHR$(30);:X
1$="":GOTO 170
230 IF LEN(X1$)>2 THEN PRINT CHR$(29);CHR$(30);:X1$="":GOTO 170
240 GOTO 180
250 IF LEN(X1$)=0 THEN GOTO 270
260 TB%=VAL(X1$):X1$=""
270 PRINT:PRINT"ENTER:  J   FOR JUSTIFICATION"
280 PRINT"ENTER:  N   FOR NON-JUSTIFICATION"
290 PRINT"ENTER:  A   TO ABORT":PRINT:PRINT"ENTER:"
300 M$=INKEY$:IF M$="" THEN GOTO 300
310 G%=ASC(M$)
320 IF G%=74 THEN QX%=0 ELSE IF G%=78 THEN QX%=1 ELSE IF G%=65 T
HEN RUN ELSE GOTO300
330 IF F%=1 THEN GOTO 1700 ELSE IF F%=2 THEN GOTO 2330
340 CLS
350 N%=0 :PRINTCHR$(128)+CHR$(13): GOTO 520
360 A$=INKEY$:IF A$="" THEN GOTO 360
370 A%=ASC(A$)
380 IFA%>32ANDA%<65GOTO550
390 IFA%>64ANDA%<91A%=A%+32:GOTO500
400 IFA%>91A%=A%-32:U%=PEEK(16416):Q%=PEEK(16417):GOTO 480
410 IF A%=27 THEN 1700
420 IF A%=32 THEN SP%=SP%+1: GOTO 550
430 IF A%=10 THEN A$=CHR$(128): GOTO 550
440 IF A%=91 THEN GOSUB 1220: GOTO 510
450 IF A%=8 THEN GOSUB 1660: GOTO 360
```

```
460 IF A%=9 THEN GOSUB 1550: GOTO 360
470 IF A%=13 THEN 510
480 IF Q%=60 THEN Q%=0 ELSE IF Q%=61 THEN Q%=256 ELSE IF Q%=62 T
HEN Q%=512 ELSE IF Q%=63 THEN Q%=768
490 POKE 15360+Q%+U%+64,183 :U%=0 :Q%=0
500 A$=CHR$(A%) : GOTO 550
510 GOSUB 1240
520  N%=N%+1:L$="":GOSUB 1610 :GOSUB 1490
530 IFN%>90THENCLS:PRINT"END OF TEXT INPUT - 90 LINES HAVE BEEN
ENTERED":FORT%=1TO1000:NEXTT%:GOTO1700
540 IF XX%=2 AND A%=13 THEN GOTO 2970 ELSE
  IF XX%=1 AND A%=13 THEN GOTO 2650 ELSE  IF XX%=1 AND A%=91THEN
  GOTO 2650
550 IF LEN(L$)<=61 THEN PRINT A$;CHR$(95);CHR$(24);
560 IF LEN(L$)=62 GOTO 360
570 L$=L$+A$
580 V%=0: GOTO 360
590 REM WORD CHANGE
600 J%=1:A%=0
610 N%=CH%:GOSUB 1490
620 FOR L%=1 TO LEN(A$(CH%))
630 J$=MID$(A$(CH%),L%,1)
640 IF J$<>" " THEN T$(J%)=T$(J%)+J$:K%=0
650 IF J$=" " THEN K%=K%+1
660 IF K%>1 THEN GOTO 680
670 IF K%=1 THEN PRINT"J  ";:J%=J%+1
680 NEXT L%
690 PRINTCHR$(29);CHR$(30);
700 FOR L%=1 TO J%
710 IF T$(L%+1)="" THEN PRINT T$(L%); ELSE PRINTT$(L%);" ";
720 NEXT L%:PRINT
730 Q%=J%
740 Z$=INKEY$:IF Z$="" THEN GOTO 740
750 A%=ASC(Z$)
760 IF DD%=Q%+1 AND T$(Q%+1)="" THEN L$=LEFT$(L$,LEN(L$)-1): FOR
I%=0 TO 20:T$(I%)="":NEXT I%:SP%=DD%-2:DD%=0:PRINT CHR$(29);CHR$(
30);:GOSUB 1240 :GOTO 2650
770 IF A%=9 THEN DD%=DD%+1:L$="":GOTO 820
780 IF A%=8 THEN DD%=DD%-1:L$="":GOTO 870
790 IF A%=91 THEN A%=0:Z$="":PRINT CHR$(176);:GOTO 910
800 IF A%=10 AND DD%=0 THEN GOTO 740 ELSE IF A%=10 THEN GOTO 113
0
810 IF A%=64 THEN GOTO 1200 ELSE 740
820 PRINT CHR$(29);CHR$(30);:L$=""
830 FOR I%=1 TO DD%
840 IF I%<>DD% THEN L$=L$+T$(I%)+" ":GOTO 860
850 IF I%=DD% THEN L$=L$+T$(I%)
860 NEXT I%:PRINT L$;:GOTO 740
870 IF DD%<=0 THEN DD%=0:PRINT CHR$(29);CHR$(30);:
     L$="":GOTO 740 ELSE PRINT CHR$(29);CHR$(30);:L$=""
880 FOR I%=1 TO DD%
890 L$=L$+T$(I%)+" "
900 NEXT I%:PRINT L$;:GOTO 740
910 Z$=INKEY$:IF Z$="" THEN GOTO 910
920 A%=ASC(Z$)
930 IF A%=8 AND LEN(TT$)>0 THEN PRINT CHR$(8);:TT$=LEFT$(TT$,LEN
(TT$)-1):GOTO 910 ELSE IF A%=8 AND LEN(TT$)=0 THEN GOTO 910
940 IF A%=96 THEN A%=128:GOTO 1120
950 IF A%<32 GOTO 910
960 IF A%=32 THEN DD%=DD%+1:Q%=Q%+1:GOTO 980
970 GOTO 1090
980 FOR I%=Q% TO DD% STEP-1
990 T$(I%)=T$(I%-1)
1000 NEXT I%:T$(DD%)=TT$:L$=""
1010 FOR I%=1 TO Q%
1020 IF I%<>Q% THEN L$=L$+T$(I%)+" ":GOTO 1040
1030 L$=L$+T$(I%)
1040 NEXT I%:TT$=""
1050 PRINT CHR$(29);CHR$(30);
1060 FOR I%=1 TO DD%
1070 PRINTT$(I%)+" ";
1080 NEXT I%:GOTO 740
1090 IF A%<65 THEN A%=A%:GOTO 1120
1100 IF A%<91 THEN A%=A%+32:GOTO 1120
1110 IF A%<128 THEN A%=A%-32
1120 Z$=CHR$(A%):PRINT Z$;:TT$=TT$+Z$:GOTO 910
1130 FOR I%=DD% TO Q%
```

```
1140 T$(I%)=T$(I%+1)
1150 NEXT I%:Q%=Q%-1:DD%=DD%-1:L$=""
1160 FOR I%=1 TO DD%
1170 L$=L$+T$(I%)+" "
1180 NEXT I%
1190 PRINTCHR$(29);CHR$(30);L$;:GOTO 740
1200 PRINTCHR$(29);CHR$(30);:L$=""
1210 FOR I%=1 TO Q%:L$=L$+T$(I%)+" ":NEXT I%:DD%=Q%+1:GOTO 760
1220 REM CENTERING TEXT
1230 Q%=LEN(L$):U%=(62-Q%)/2:L$=STRING$(U%,128)+L$:RETURN
1240 REM JUSTIFY TEXT
1250 J%=0:W%=0:IF LEN(L$)<55 OR LEN(L$)>=64 OR QX%=1 THEN GOTO
 1430 ELSE PRINT CHR$(29);CHR$(30);:FOR I%=1 TO LEN(L$)
1260 A$=(MID$(L$,I%,1))
1270 IF A$<>" "THEN T$(J%)=T$(J%)+A$
1280 IF A$=" " THEN J%=J%+1
1290 NEXT I%:I%=0:J%=0:A$=""
1300 X=(64+SP%-LEN(L$))/SP%:X=INT(X):W%=(64+SP%)-((X*SP%)+(LEN(L
$)))
1310 FOR J%=0 TO SP%-1
1320 T$(J%)=T$(J%)+STRING$(X,32):PRINT"J  ";
1330 NEXT J%
1340 L$=""
1350 FOR J%=0 TO SP%
1360 IF SP%>=0 AND W%>0 THEN T$(J%)=T$(J%)+CHR$(128)
1370 L$=L$+T$(J%)
1380 SP%=SP%-1
1390 W%=W%-1
1400 NEXT J%:PRINT
1410 FOR J%=0 TO 20:T$(J%)="":NEXTJ%:SP%=0
1420 L$=LEFT$(L$,64):PRINTCHR$(29);CHR$(30);CHR$(27);L$;: GOTO14
40
1430 PRINT CHR$(29);CHR$(30);:
     IF LEN(L$)=0 THEN PRINTCHR$(128)+CHR$(13); ELSE IF LEN(L$)<
64 THEN PRINT L$+CHR$(13); ELSE IF LEN(L$)=64 THEN L$=LEFT$(L$,64
):PRINT L$; ELSE IF LEN(L$)>64 THEN L$=LEFT$(L$,63):PRINT L$;CHR$
(43);
1440 IF LEN(L$)=0 THEN A$(N%)=" " ELSE A$(N%)=L$
1450 L$="":SP%=0:X=0:J%=0:A$="":W%=0
1460 RETURN
1470 PRINT"ERROR=#";ERR/2+1,"ERROR LINE=";ERL
1480 FOR T%= 1 TO 1000:NEXTT%:RESUME 1700
1490 REM LINE COUNTER
1500 I$=STR$(N%) : CL%=LEN(I$)
1510 IF CL% =2 THEN H%=48:L%=ASC(RIGHT$(I$,1))
1520 IF CL% =3 THEN H%=ASC(MID$(I$,2,1)):L%=ASC(RIGHT$(I$,1))
1530 POKE 15413,91:POKE 15414,32:POKE 15415,76:POKE 15416,73:POK
E 15417,78:POKE 15418,69:POKE 15419,32:POKE 15420,H%:POKE 15421,L
%:POKE 15422,32:POKE 15423,91
1540 RETURN
1550 REM INDENTING
1560 IF TB%=0 THEN RETURN
1570 IF TB%>64 THEN TB%=64
1580 TB%=TB%-1:PRINTSTRING$(TB%,32);CHR$(95);CHR$(24);
1590 L$=STRING$(TB%,128)+L$
1600 TB%=TB%+1:RETURN
1610 REM PRINT INDENT POSITION
1620 PRINTCHR$(26);
1630 IF TB%=0 GOTO 1650
1640 PRINTSTRING$(TB%-1,95);CHR$(91);
1650 PRINTSTRING$(62-TB%,95);STRING$(2,128);:PRINTCHR$(27);CHR$(
27);CHR$(29);CHR$(30);:RETURN
1660 REM BACK SPACING
1670 IF LEN(L$)=0 THEN RETURN ELSE
     IF RIGHT$(L$,1)=" " THEN SP%=SP%-1
1680 PRINTCHR$(8);CHR$(30);:IF LEN(L$)<63 THEN PRINT CHR$(95);
     CHR$(24);:L$=LEFT$(L$,LEN(L$)-1):U%=PEEK(16416):Q%=PEEK(164
17):IFQ%=60THENQ%=0ELSEIFQ%=61THENQ%=256ELSEIFQ%=62THENQ%=512ELSE
IFQ%=63THENQ%=768
1690 POKE 15360+Q%+U%+64,95: U%=0: Q%=0:RETURN
1700 CLS:F%=0
1710 PRINTTAB(12)"ENTER:  1  TO PRINT DRAFT DOCUMENT"
1720 PRINTTAB(12)"ENTER:  2  TO PRINT COMPLETED TEXT":PRINTTAB(1
2) STRING$(34,45)
1730 PRINTTAB(12)"ENTER:  3  TO RECORD TO TAPE"
1740 PRINTTAB(12)"ENTER:  4  TO LOAD FROM TAPE":PRINTTAB(12) STR
ING$(34,45)
```

```
1750 PRINTTAB(12)"ENTER:  5  TO REVIEW DATA":PRINTTAB(12) STRING
$(34,45)
1760 PRINTTAB(12)"ENTER:  6  TO CHANGE DATA":PRINTTAB(12) STRING
$(34,45)
1770 PRINTTAB(12)"ENTER:  7  TO ADD TO THE DOCUMENT":PRINTTAB(12
) STRING$(34,45)
1780 PRINTTAB(12)"ENTER:  8  TO INSERT NEW LINE"
1790 PRINTTAB(12)"ENTER:  9  TO DELETE A LINE"
1800 PRINTTAB(12) STRING$(34,45):PRINTTAB(12)"ENTER: CHOICE  ";:
INPUT Y$
1810  IF Y$="INDENT" THEN F%=1: CLS: PRINT: GOTO 150
1820 IF LEN(Y$)>1 THEN GOTO 1700
1830 Y%=VAL(Y$): IF Y%<1 OR Y%>9 THHEN GOTO 1700
1840 ON Y% GOTO 1850,1960,2050,2280,2710,2510,2030,2820,3020
1850 CLS:Z$="":PRINT"MAKE PRINTER READY "
1860 PRINT:PRINT"ELSE  ENTER: 'A' TO ABORT ";:INPUT Z$
1870 IF Z$<>"" THEN GOTO 1700
1880 LPRINT" INDENTATION =";TB%;
1890 IF QX%=0 THEN LPRINT"  JUSTIFY" ELSE LPRINT
     "  NON-JUSTIFY"
1900 LPRINT" "
1910 FOR I%=1 TO N%
1920 LPRINTCHR$(29);CHR$(27);CHR$(56);CHR$(27);CHR$(66);I%;TAB(2
5);A$(I%):LPRINT" ":IF A$(I%)=""THEN LPRINT " "
1930 IF I%=40 OR I%=81 THEN GOSUB 2490
1940 NEXT I%:GOTO 1700
1950 A$="":A%=0:L$="":CLS:N%=N%+1:GOSUB 1610:GOSUB 1490:GOTO 360
1960 CLS:Z$="":PRINT"MAKE PRINTER READY "
1970 PRINT:PRINT"ELSE ENTER: 'A'  TO ABORT ";:INPUT Z$
1980 IF Z$<>"" THEN GOTO 1700
1990 FOR I%=1 TO N%
2000 IF I%=59 THEN GOSUB 2490
2010 LPRINTCHR$(27);CHR$(66);CHR$(27);CHR$(54);CHR$(30); A$(I%):
IF A$(I%)="" THEN LPRINT " "
2020 NEXT I%:GOTO 1700
2030 CLS:L$="":V%=3:IF V%>=N% THEN V%=1
     ELSE V%=N%-3
2040 CLS:FOR I%=V% TO N%:PRINTA$(I%):NEXT I%:GOSUB 1610:GOSUB 14
90:
     GOTO 360
2050 REM RECORD DATA
2060 CLS
2070 FOR I%=1 TO 90
2080 IF A$(I%)="" THEN GOTO 2090 ELSE N%=I%
2090 NEXT I%:PRINT"THERE ARE ";N%;" LINES":PRINT
2100 Z$="":PRINT"MAKE TAPE RECORDER READY FOR RECORDING "
2110 PRINT:PRINT"ELSE ENTER: 'A'  TO ABORT ";:INPUT Z$
2120 IF Z$<>"" THEN GOTO 1700
2130 CLS:PRINTTAB(15);"THE TAPE RECORDER IS RUNNING":PRINT:PRINT
"IT WILL PAUSE, HOWEVER, WHILE IT CONVERTS COMMAS AND COLONS
TO GRAPHIC CHARACTERS"
2140 PRINT #-1,N%
2150 FOR I%=1 TO N%
2160 FOR J%=1 TO LEN(A$(I%))
2170 K$=MID$(A$(I%),J%,1)
2180 IF K$="," THEN KK$=KK$+CHR$(168):GOTO 2210
2190 IF K$=":" THEN KK$=KK$+CHR$(162):GOTO 2210
2200 KK$=KK$+K$
2210 NEXT J%:A$(I%)=KK$ :KK$=""
2220 NEXT I%
2230 FOR I%=1TO N% STEP 3
2240 PRINT#-1,A$(I%),A$(I%+1),A$(I%+2)
2250 PRINTA$(I%):PRINTA$(I%+1):PRINTA$(I%+2)
2260 IF A$(I%)="" OR A$(I%+1)="" OR A$(I%+2)="" THEN PRINT CHR$(
128)
2270 NEXT I%:GOTO2400
2280 REM READ DATA
2290 CLS:Z$="":PRINT TAB(25);"LOAD FROM TAPE":PRINT:PRINT"MAKE T
APE RECORDER READY FOR PLAYING"
2300 PRINT:PRINT"ELSE ENTER: 'A'  TO ABORT  ";:INPUT Z$
2310 IF Z$<>"" THEN GOTO 1700
2320 PRINT:F%=2:GOTO 150
2330 CLS:PRINTTAB(15);"THE TAPE RECORDER IS RUNNING"
2340 PRINT
2350 INPUT #-1, N%: PRINT"THERE ARE ";N%;" LINES":PRINT
2360 FOR I%=1 TO N% STEP 3
2370 INPUT #-1, A$(I%),A$(I%+1),A$(I%+2)
```

```
2380 PRINTA$(I%):PRINTA$(I%+1):PRINTA$(I%+2)
2390 NEXT I%
2400 CLS:PRINT"THE PROGRAM IS NOW REPLACING THE GRAPHIC CHARACTE
RS              WITH COMMAS AND COLONS."
2410 FOR I% =1 TO N%
2420 FOR J%=1 TO LEN(A$(I%))
2430 K$=MID$(A$(I%),J%,1)
2440 IF K$= CHR$(168)  THEN KK$=KK$+",":GOTO 2470
2450 IF K$= CHR$(162) THEN KK$=KK$+":":GOTO 2470
2460 KK$=KK$+K$
2470 NEXT J%:A$(I%)=KK$:KK$=""
2480 NEXT I%:GOTO 1700
2490 CLS:PRINT"PAGE FULL, INSERT ANOTHER SHEET IN PRINTER":PRINT
:PRINTTROFFENTER  'CONT'   TO PRINT BALANCE OF TEXT":PRINT:STOP
2500 RETURN
2510 REM CHANGE DATA
2520 L$="":FF%=N%+1:F$=STR$(N%+1)
2530 CLS:Z$=""
2540 PRINTCHR$(29);CHR$(30);"ENTER LINE TO BE CHANGED ";:INPUT C
H%
2550 IF N%=0 THEN GOTO 1700
2560 PRINT:PRINTA$(CH%)
2570 IF CH%<1 OR CH%>N% THEN GOTO 2540
2580 PRINT:PRINT"ENTER:  L  FOR LINE CHANGE":
           PRINT"ENTER:  W  FOR WORD CHANGE":
           PRINT"ENTER:  A  TO ABORT":PRINT
2590 M$=INKEY$:IF M$="" GOTO 2590
2600 G%=ASC(M$)
2610 IF G%=87 THEN WD%=1 ELSE IF G%=76 THEN WD%=0 ELSE GOTO 1700
2620 PRINTA$(CH%):PRINT:XX%=1
2630 IF WD%=1 THEN GOTO 590
2640 N%=CH%:GOSUB 1610:GOSUB 1490:GOTO 360
2650 PRINT:PRINT"FOR ANOTHER CHANGE  'C'  ELSE  'ENTER'";
2660 Z1$=INKEY$:IF Z1$="" THEN GOTO 2660
2670 Z%=ASC(Z1$)
2680 IF Z%=13 THEN XX%=0:N%=FF%-1:GOTO 1700
2690 IF Z%=67 THEN N%=FF%: GOTO 2530
2700 IF Z%<>13 OR Z%<>67 THEN GOTO 2650
2710 REM REVIEW DATA
2720 CLS:PRINTTAB(20);"REVIEW DATA":PRINT
2730 PRINT"ENTER 'A' TO ABORT  ELSE  'ENTER' "
2740 Z$=INKEY$:IF Z$="" THEN GOTO 2740
2750 Z%=ASC(Z$):IF Z%<>13 THEN GOTO 1700
2760 FOR Q%=1 TO N%
2770 PRINTA$(Q%)
2780 N%=Q%:GOSUB 1490
2790 Z$=INKEY$:IF Z$="" THEN GOTO 2790
2800 Z%=ASC(Z$):IF Z%<>13 THEN GOTO 1700
2810 NEXT Q%:GOTO 1700
2820 REM INSERT LINES
2830 CLS:Z$="":PRINT TAB(25);"INSERT LINE":PRINT
2840 CLS:PRINT"ENTER STARTING NUMBER OF INSERT  ";:INPUT E%
2850 PRINTA$(E%-1):PRINT"INSERT":PRINTA$(E%):PRINT
2860 PRINT"INSERT LINE  ELSE  'A' TO ABORT  ";:INPUT Z$
2870 IF Z$<>"" THEN GOTO 1700
2880 IF E%<1 OR E%>N% THEN GOTO 2840
2890 R%=0
2900 FOR I%=E% TO N%:R%=R%+1
2910 B$(R%)=A$(I%):A$(I%)=""
2920 NEXT I%
2930 L$="":XX%=2
2940 PRINT:PRINT"ENTER LINE TO BE ADDED  ";:PRINT:PRINT
2950 N%=E%
2960 GOSUB 1610:GOSUB 1490:GOTO 360
2970 FOR I%=(E%+1) TO (R%+E%):K%=K%+1
2980 A$(I%)=B$(K%):B$(K%)=""
2990 NEXT I%
3000 N%=R%+E%
3010 R%=0:K%=0:XX%=0:GOTO 1700
3020 REM DELETE LINE
3030 CLS:Z$="":Y%=0:X%=0:PRINTTAB(25);"DELETE LINE":PRINT
3040 PRINT"ENTER LINE NUMBER TO BE DELETED  ";:INPUT Y%:
     PRINT:PRINT A$(Y%):PRINT
3050 Z$="":PRINT"DELETE  LINE  ELSE  'A'  TO ABORT  ";:INPUT Z$
3060 IF Z$<>"" THEN GOTO 1700
3070 IF N%=0 THEN GOTO 1700
3080 IF Y%<1 OR Y%>N% THEN GOTO 3030
```

```
3090 FOR I%=1 TO N%+1
3100 IF Y%=I% THEN X%=1
3110 IF X%=1 THEN A$(I%)=A$(I%+1)
3120 NEXT I%:N%=N%-1:GOTO 1700
3130 END
```

***** THE TOWERS OF HANOI by M. Byrne *****

The 'Towers of Hanoi' is played with a number of discs of differing sizes and three pegs on which the discs may be stacked. Initially the discs are arranged on one peg in order of decreasing size. The object is to transfer them (in the least number of moves) to another of the pegs so they are once again arranged in order of decreasing size. However, you may only move one disc at a time and it must not be placed on a smaller disc.

This program allows you to select the number of discs (up to a maximum of 10) and whether you or the computer will solve the problem.

The main variables used are:

N(3)	holds the number of discs on each peg.
P1(), P2(), P3()	are peg arrays. Each element holds the size of the disc at that position.
SK()	the parameter stack.
SC	the source peg.
DN	the destination peg.
AL	the alternate peg.
NM	the move counter.
SP	the stack pointer.
B	the base pointer for the stack elements of the previously invoked procedure.
CT	the number of discs.
S	the amount of delay (for automatic operation).
F	the peg the disc is moved from.
T	the peg the disc is moved to.
T1, T2	used to test whether the move is legal.
M	is the minimum number of moves.
TD	the top disk of the peg we are moving from.

Probably the feature which will cause most confusion is the use of recursion. It appears to be a not-too-well-known fact that BASIC will allow recursion (on most machines anyway). Recursion is a means whereby a procedure or subroutine may call itself. This involves the use of a stack (which BASIC provides) for storing the return address and another stack (which must be set up explicitly) if parameters are to be passed to the called procedure.

The parameter stack is only really used when automatic operation is required as manual operation merely involves asking for moves and checking their validity. The program is organised as follows:

Lines 7-40	initialisation
Lines 50-110	obtain the number of discs and determine the mode of operation.
Lines 120-190	contain the automatic operation logic sets up the stack pointers and initialises the stack.
Lines 200-400	are the manual operation logic which asks for moves and checks their validity.
Lines 460-490	draw the discs initially.
Lines 500-700	are responsible for shifting the discs. The delay routine is Line 505.
Lines 1000-1410	is the main recursive procedure which is only executed in automatic operation. It decides what the new source and destination pegs should be, places these parameters on the stack SK() and calls itself.
Lines 1500-1660	draws the pegs.
Lines 1700-1780	is the graphics routine for erasing the disc on the old peg and redrawing it on the new peg.
Lines 2000-2070	print the start up message.
Lines 2500-2980	contain the instructions.

The current top-of-stack is indicated by SP. When used by the main procedure (Lines 1000-), the top stack element holds the number of the alternate peg for this move, top of stack - 1 holds the number of the destrination peg for this move, top of stack - 2 holds the number of the source peg for this move while top of stack - 3 indicates the number of discs still to be moved.

When used by the shift procedures (Lines 500-), the top of stack contains the number of the destination peg and top of stack - 1 contains the number of the source peg.

Because the parameter stack must be explicitly set up and manipulated in BASIC, the algorithm tends to become somewhat obscured. Also if it is the first time you have encountered recursion, BASIC is not the ideal language to learn about it. For those of you who are interested in finding out more, the tree-diagram for the algorithm used in this program is given in the book 'An Introduction to Problem Solving Using Pascal' by Kenneth Bowles.

```
 40 CLEAR 93
 50 CLS:GOSUB 1430:GOSUB 1510:MAX=10
 60 DIM N(3),P1(MAX),P2(MAX),P3(MAX),SK(10*MAX)
 70 SC=1:AL=2:DN=3:CT=0:P1(0)=99:P2(0)=99:P3(0)=99
 80 NM=0:J=0:I=0:SP=0:B=0:CLS
 90 INPUT"HOW MANY DISCS WOULD YOU LIKE";CT
100 IF CT>MAX PRINT"THAT WILL TAKE SOME TIME AND FIRST YOU WILL
HAVE TO  CHANGE LINE 50":END
110 IF CT<=0 PRINT"SORRY, THAT'S NOT POSSIBLE":GOTO 90
120 IF CT<=3 PRINT"YOU HAVE NO SENSE OF ADVENTURE - BUT I SUPPOS
E
YOU HAVE TO START SOMEWHERE"
130 N(1)=CT:N(2)=0:N(3)=0
140 IF AN$="YES" OR AN$="NO" THEN PRINT"TYPE 'A' IF YOU WANT ME
TO DO ALL THE WORK (I.E. AUTOMATIC
OPERATION)  OTHERWISE TYPE 'M' FOR MANUAL OPERATION":PRINT"WHICH
 WOULD YOU LIKE ";
150 INPUT"AUTOMATIC OR MANUAL (A OR M)";MODE$
160 IF MODE$<>"A" GOTO 280
170 INPUT"HOW FAST (0 IS FASTEST, 10 IS SLOWEST)";S
180 CLS:GOSUB1160
190 GOSUB540:SP=4
200 SK(SP-3)=CT:SK(SP-2)=SC:SKK(SP-1)=DN:SK(SP)=AL
210 B=SP:SP=SP+4
220 SK(SP-3)=SK(B-3):SK(SP-2)=SK(B-2):SK(SP-1)=SK(B-1):SK(SP)=SK
(B)
230 IF CT<>1 THEN GOTO 260
240 SP=SP+2:SK(SP-1)=1:SK(SP)=3:GOSUB 590
250 GOTO270
260 GOSUB 820
270 PRINT@832,"";:GOTO510
280 CLS:GOSUB 1160:S=0
290 GOSUB 540
300 PRINT@832,"                                                  ";
310 PRINT@832,"FROM PEG";:INPUT F
320 PRINT@860,"TO PEG";:INPUT T
330 IF F<1 OR T<1 OR F>3 OR T>3 PRINT"NO SUCH PEG":GOTO 300
340 IF F=T PRINT"NOT ALLOWED":GOTO 300
350 IF N(F)<=0PRINT"THERE ARE NO DISCS ON PEG ";F:GOTO300
360 I=N(F):J=N(T)
370 ON F GOTO 380,410,430
380 T1=P1(I)
390 IF T=2 THEN T2=P2(J):ELSE T2=P3(J)
400 GOTO 440
410 T1=P2(I):IF T=1 THEN T2=P1(J):ELSE T2=P3(J)
420 GOTO 440
430 T1=P3(I):IF T=2 THEN T2=P2(J):ELSE T2=P1(J)
440 IF T1>T2 PRINT"NOT ALLOWED":GOTO300
450 SP=SP+2:SK(SP-1)=F:SK(SP)=T
460 GOSUB 590
470 IF N(1)=0 AND N(2)=0 AND N(3)=CT THEN GOTO 480:ELSE GOTO 300
480 M=2[CT-1:PRINT@832,"
   ":PRINT@832,"";
490 IF NM=M THENPRINT"CONGRATULATIONS! YOU DID IT" ELSE IF NM-M<
=M*0.15 THENPRINT"NOT BAD AT ALL"
500 PRINT"YOU TRANSFERRED ALL THE DISCS IN ";NM;" MOVES
510 PRINT"CARE FOR ANOTHER GAME (YES OR NO)";
520 INPUT AN$
530 IF AN$="N" OR AN$="NO"THEN CLS:END ELSE GOTO80
540 J=CT
550 FOR I=1 TO CT
560 P1(I)=J:J=J-1:NEXT I
570 RETURN
580 '=============  SHIFT DISCS
590 NM=NM+1:SE=SK(SP-1):DE=SK(SP)
600 FOR I=1 TO 30*S:NEXT I
610 I=N(SE):Y=33-(I*2)
620 IF SE<>1 THEN 640
630 TD=P1(I):X=20:GOTO 670
640 IF SE<>2 THEN 660
650 TD=P2(I):X=52:GOTO 670
660 TD=P3(I):X=84
670 P=2:SIZE=TD
680 GOSUB 1340
690 I=N(DE)+1:Y=33-(I*2)
700 IF DE<>1 THEN 720
710 P1(I)=TD:X=20:GOTO 750
```

```
720 IF DE<>2 THEN 740
730 P2(I)=TD:X=52:GOTO 750
740 P3(I)=TD:X=84
750 P=1
760 GOSUB 1340
770 N(SE)=N(SE)-1
780 N(DE)=N(DE)+1
790 SP=SP-2
800 PRINT@23,"MOVE ";NM
810 RETURN
820 '=============  HANOI(RECURSIVE PROCEDURE)
830 IF SK(SP-3)>2 THEN GOTO 840 ELSE GOTO 1010
840 B=SP:SP=SP+4
850 SK(SP-3)=SK(B-3)-1
860 SK(SP-2)=SK(B-2)
870 SK(SP-1)=SK(B)
880 SK(SP)=SK(B-1)
890 GOSUB 820
900 B=SP:SP=SP+2
910 SK(SP-1)=SK(B-2)
920 SK(SP)=SK(B-1)
930 GOSUB 590
940 B=SP:SP=SP+4
950 SK(SP-3)=SK(B-3)-1
960 SK(SP-2)=SK(B)
970 SK(SP-1)=SK(B-1)
980 SK(SP)=SK(B-2)
990 GOSUB 820
1000 GOTO 1130
1010 B=SP:SP=SP+2
1020 SK(SP-1)=SK(B-2)
1030 SK(SP)=SK(B)
1040 GOSUB 590
1050 B=SP:SP=SP+2
1060 SK(SP-1)=SK(B-2)
1070 SK(SP)=SK(B-1)
1080 GOSUB 590
1090 B=SP:SP=SP+2
1100 SK(SP-1)=SK(B)
1110 SK(SP)=SK(B-1)
1120 GOSUB 590
1130 SP=SP-4
1140 RETURN
1150 '=============  DRAWPEGS
1160 CLS:' PROC DRAWPEGS
1170 PRINT@256,"";:A$=CHR$(191)
1180 FOR I=1 TO 7
1190 PRINT"          ";A$;"                ";A$;"                "
;A$
1200 NEXT I
1210 PRINT"     ";
1220 FOR I=1 TO 44:PRINT A$;:NEXT I
1230 PRINT:PRINT"          1                2                3"
1240 Y=31:X=20
1250 FOR I=CT TO 1 STEP -1
1260    FOR J=1 TO I
1270            SET(X-J,Y)
1280            SET(X+J+2,Y)
1290    NEXT J
1300    Y=Y-2
1310 NEXT I
1320 RETURN
1330 '=============  SWITCH PROCEDURE
1340 IF P=2 THEN 1390
1350 FOR J=1 TO SIZE
1360    SET(X-J,Y):SET(X+J+2,Y)
1370 NEXT J
1380 GOTO 1420
1390 FOR J=1 TO SIZE
1400    RESET(X-J,Y):RESET(X+J+2,Y)
1410 NEXT J
1420 RETURN
1430 D$=STRING$(63,"*")
1440 PRINT@256,D$
1450 PRINT D$:PRINT
1460 PRINT@463,"T H E    T O W E R S"
1470 PRINT@542,"O F    H A N O I":PRINT
```

```
1480 PRINT D$
1490 PRINT D$
1500 RETURN
1510 FORI=1 TO 1000:NEXT I
1520 PRINT@896,"DO YOU WANT INSTRUCTIONS (YES OR NO)";:INPUT AN$
1530 IF AN$<>"Y" AND AN$<>"YES" THEN RETURN
1540 CLS
1550 PRINT@23,"INSTRUCTIONS"
1560 PRINT@87,"------------":PRINT
1570 PRINT"    THERE ARE THREE PEGS.  ON ONE OF THESE IS ARRANGE
D,
1580 PRINT"IN ORDER OF DECREASING SIZE, A NUMBER OF DISCS.":PRIN
T
1590 PRINT"    THE OBJECT IS TO MOVE ALL THE DISCS FROM THE LEFT
MOST"
1600 PRINT"PEG (1) TO THE RIGHTMOST PEG (3), SUBJECT TO THE FOLL
OWING"
1610 PRINT"CONSTRAINTS :-":PRINT
1620 PRINT"    1. ONLY ONE DISC MAY BE MOVED AT A TIME"
1630 PRINT"    2. A DISC MAY NOT BE PLACED ON TOP OF A DISC WHIC
H"
1640 PRINT"IS SMALLER"
1650 GOSUB 2000
1660 PRINT@23,"PLAYING HANOI"
1670 PRINT@87,"-------------":PRINT
1680 PRINT"    THERE ARE TWO MODES OF PLAY.  THESE ARE <A>UTOMAT
IC"
1690 PRINT"AND <M>ANUAL AND YOU WILL BE ASKED TO SELECT ONE.":PR
INT
1700 PRINT"AUTOMATIC OPERATION"
1710 PRINT"    THE COMPUTER WILL ASK YOU HOW MANY DISCS YOU WOUL
D LIKE"
1720 PRINT"MOVED AND HOW FAST YOU WOULD LIKE THEM MOVED.  IT WIL
L"
1730 PRINT"THEN PROCEED TO TRANSFER THE DISCS FROM PEG 1 TO PEG
3"
1740 PRINT"USING PEG 2 AS AN INTERMEDIATE."
1750 GOSUB 2000
1760 PRINT@23,"PLAYING HANOI"
1770 PRINT@87,"-------------":PRINT
1780 PRINT"MANUAL OPERATION"
1790 PRINT"    HERE YOU WILL BE ASKED HOW MANY DISCS YOU WOULD L
IKE"
1800 PRINT"TO MOVE.  THEN YOU WILL BE PROMPTED TO TYPE THE NUMBE
R OF"
1810 PRINT"THE PEG THE DISC IS TO BE TAKEN OFF AND THE NUMBER OF
 THE"
1820 PRINT"PEG THE DISC IS TO BE PLACED ON.  WHEN YOU HAVE SUCCE
SSFULLY"
1830 PRINT"TRANSFERRED ALL DISCS YOU WILL BE TOLD HOW MANY MOVES
"
1840 PRINT"IT TOOK."
1850 GOSUB2000
1860 PRINT@23,"SUGGESTION"
1870 PRINT@87,"----------":PRINT
1880 PRINT"    IF YOU ARE UNSURE OF THE GAME TRY AUTOMATIC OPERA
TION"
1890 PRINT"WITH ABOUT FOUR DISCS AND LOW SPEED (E.G. 10).  THEN
SIT"
1900 PRINT"BACK AND LET THE COMPUTER DO ALL THE WORK - AFTER ALL
 THAT"
1910 PRINT"IS WHAT IT IS THERE FOR.":PRINT
1920 PRINT"    FOR A GIVEN NUMBER OF DISCS, N, THE SMALLEST NUMB
ER OF"
1930 PRINT"MOVES REQUIRED IS GIVEN BY :-"
1940 PRINT"                M = 2↑N-1
1950 PRINT"SO FOR 3 DISCS THAT'S 7 MOVES, 4 DISCS 15 MOVES UP TO
"
1960 PRINT"THE MAXIMUM OF 10 DISCS WHICH REQUUIRES 1023 MOVES.":
PRINT
1970 PRINT"    HAVE FUN!"
1980 GOSUB 2000
1990 RETURN
2000 PRINT@960,"PRESS <NEWLINE> WHEN READY";
2010 INPUT D$:CLS
2020 RETURN
```

***** NEXT MONTH'S ISSUE *****

Next month's issue will contain at least the following programs plus the usual features and articles. There will also be an index for the past 12 issues.

** TRIANGLE SOLUTIONS LI/4K **

This Level I program has been designed to calculate unknown sides, angles and the area of any triangle, useful for anyone who needs to calculate areas for concrete pours for example.

** STEEPLECHASE LII/16K **

We have had horse races before but never one like this - make your horse jump at the right time or off you come. Realistic graphics include the course, jumps, you (the jockey), a very well-animated horse and even an ambulance to take you to hospital.

** BASIC AND LABELS 32K/DISK **

When you are writing a complicated program wouldn't it make life easier if you could type - GOSUB FRED and further on in the program is a subroutine called FRED or, for that matter, any name that suits you. Well, after next month, that's just what you will be able to do.

** CURVILINEAR REGRESSION AND POLYNOMINAL REGRESSION LII/4K **

These are the last two programs in the current series of scientific programs. Curvilinear regression is used to compute the curvilinear relationship between two sets of data and Polynominal regression uses a mathematical model series which becomes increasingly complex with the addition of further degrees of Polynomiality.

** 3-D MAZE LII/16K **

This excellent program displays a maze on your screen as viewed from inside. You can turn or move by pressing a single key. As you move, so the view is changed. Each time you move it costs you points. The object is to get out of the maze with as many points as possible. You can even call up a map of the maze that shows your current position in relation to the exit, but it costs you a LOT of points to do that. Each time you run the program you get a different maze.

APPLICATION FOR PUBLICATION OF A PROGRAM IN MICRO-80

Date .....

Tick where appropriate

To **MICRO-80**
Please consider the enclosed program for . . .

(i) Publication in MICRO-80 .....

(ii) Publication on disk or cassette only .....

(iii) Both .....

Name .....

Address .....

.....

.....Postcode .....

*** CHECK LIST ***

Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level I, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes separated from the accompanying literature.

Ensure that you supply adequate instructions, notes on what the program does and how it does it, etc.

For system tapes, the start, end, and entry points, etc.

The changes or improvements that you think may improve it.

Please package securely — padabags are suggested — and enclose stamps or postage if you want your cassette or disk returned.

***** CASSETTE EDITION INDEX *****

The cassette edition of MICRO-80 contains all the software listed each month, on cassette. All cassette subscribers need do is CLOAD and RUN the programs. Level II programs are recorded on side 1 of the cassette. Level I programs are recorded on side 2. Level I programs are not compatible with the System 80. All programs are recorded twice in succession. Note, System 80 computers have had different tape-counters fitted at different times. The approximate start positions shown are correct for the very early System 80 without the volume control or level meter. They are probably incorrect for later machines. The rates for a cassette subscription are printed on the inside front cover of each issue of the magazine.

The disk edition contains all those programs which can be executed from disk, including Level I programs. Level I disk programs are saved in the NEWDOS format. Users require the Level I/CMD utility supplied with NEWDOS + or NEWDOS 80 version 1.0 to run them.

	TYPE	I.D.	DISK FILESPEC	APPROX. START POSITION CTR-41	CTR-80	SYSTEM-80
SIDE ONE						
MULTIPLE REGRESSION	LII/4K	M	MULTREG/BAS	15	10	10
"	"	"	"	50	34	36
ATOMIC TABLES	LII/16K	A	ATOMIC/BAS	85	57	60
"	"	"	"	128	86	90
TEXT TYPER	LII/16K	T	TEXTYPER/BAS	170	114	120
"	"	"	"	237	160	168
TOWERS OF HANOI	LII/16K	B	TOWERS/BAS	300	202	212
SIDE TWO						
TOWERS OF HANOI	LII/16K	B	TOWERS/BAS	15	10	10
ATTACK	LII/16K	C	ATTACK/BAS	70	47	50
"	"	"	"	108	73	77
BASIC LINE VALIDATOR	EDTASM	VALID	VALID/EDT	145	98	103
"	"	"	"	162	109	115
"	SYSTEM	"	VALID/CMD *	177	119	125
"	"	"	"	180	122	128
TIC TAC TOE	LI/16K	-	TICKTACTO/LV1	185	125	-
"	"	-	"	248	167	-

A SPECIAL OFFER!

TO SUBSCRIBERS OF MICRO 80

FOR THE FIRST TIME IN AUSTRALIA!
GENUINE TANDY TRS–80 MICRO-COMPUTERS, HARDWARE AND SOFTWARE AT

GREAT DISCOUNT PRICES!!!

HOW TO ORDER –

1. SELECT ITEMS FROM 1980 TANDY CATALOGUE
2. DEDUCT 10% FROM ADVERTISED PRICES
3. POST US YOUR ORDER STATING DESCRIPTION CAT. No AND A CHEQUE OR MONEY ORDER.

WE WILL –

1. ATTEND TO YOUR ORDER WITHIN 7 DAYS *
2. SUPPLY GOODS SELECTED FREIGHT FREE !
3. SEND ADVERTISING REGULARLY TO KEEP YOU INFORMED OF CURRENT SPECIALS !

* *subject to availability*

to **CONQUEST ELECTRONICS** Pty. Ltd.
212 Katoomba St. KATOOMBA 2780

Please supply –

QTY.	CAT NO	DESC.	ADV. PRICE

SUB TOTAL _____
LESS 10% _____
FIND CHEQUE FOR **TOTAL**

SEND **FREIGHT FREE** TO

NAME .....

ADDRESS .....

.....P/ Code .....

TANDY ELECTRONICS DEALER

CONQUEST ELECTRONICS PTY. LTD.

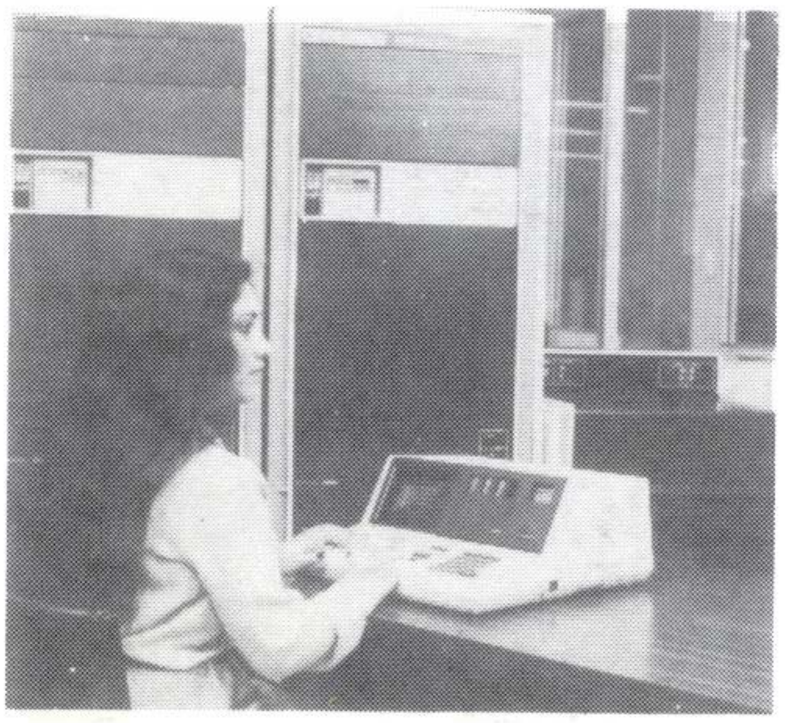
212 KATOOMBA ST KATOOMBA N.S.W. 2780 PHONE (047) 82 2491

MICRO-80

LET THE

AUSTRALIAN SOURCE

BRING MAINFRAME POWER INTO YOUR HOME OR OFFICE



WE ARE PROUD TO ANNOUNCE THE LAUNCHING OF AUSTRALIA'S FIRST MICROCOMPUTER INFORMATION UTILITY

The AUSTRALIAN SOURCE is Australia's first microcomputer information utility, aimed at giving the average microcomputer user access to the computer data banks and also massive storage space previously only available to large Mainframe installations. The era of home computers has now officially begun in Australia, now that the AUSTRALIAN SOURCE is here; as now, any small businessman or student can have on his desk for an extremely small cost, a system that has the level of storage power and access to data banks for which only a few years ago, government departments and large private users were paying millions of dollars.

The AUSTRALIAN SOURCE can be accessed through a number of 'approved' personal computers, and terminals, by use of an accoustic coupler or modem.

Users will receive a number of benefits which will include:

INFORMATION SERVICES

Members will have access to a number of information sources which will include the latest news, sports results, financial reports, and farm information.

NATIONWIDE ELECTRONIC MAIL

Members will be able, also, to communicate nationwide with other AUSTRALIAN SOURCE members through our electronic mail system.

THE AUSTRALIAN SOURCE RECOMMENDS 3M DISKETTES

SOFTWARE BANK

Users will have access to a myriad of computer programs that will include entertainment, educational aids, programming and diagnostic tools, and financial applications.

MAINFRAME POWER

Users will have the capability to make use of the Mainframe's huge storage capacity by using any of our large programs, or storing your large programs on our system.

SHOP AT HOME

You can take advantage of our 'shopping by computer' system to get the best prices on a number of popular consumer items.

EXTREMELY EASY TO USE

You do not have to be a computer programmer to make use of the AUSTRALIAN SOURCE. All of the instructions are in everyday English, so that even the younger members of your family will be able to operate the system.

LOW COST

While services like these used to cost tens of thousands of dollars to the government departments and large corporations who used them, they are now available to you for less than the cost of a packet of cigarettes a day for the 'average' user.

You can join the AUSTRALIAN SOURCE by paying a one–time joining fee (normally $100) and a small hourly user charge (normally $10 an hour 8 a.m. – 6 p.m. and $4.50 an hour 6 p.m. – 8 a.m.).

SPECIAL CHARTER MEMBER OFFER

As a special promotion, the joining fee before the 1st February, 1982, is only $60 and this also entitles you to 20% off all list user charges in the future – a great deal for the money.

TO JOIN THE AUSTRALIAN SOURCE FAMILY, AND TAKE ADVANTAGE OF THE EXCITING SERVICES WE WILL BE OFFERING, PLEASE COMPLETE THE COUPON BELOW:

MEMBERSHIP REQUEST

Yes, I think the idea of the AUSTRALIAN SOURCE is great! I enclose $60, so please send me my user manual and password.

☐ *I have a computer, Brand.....*

Model.....

☐ *Please send me information on low cost equipment packages I can use to take advantage of the AUSTRALIAN SOURCE'S services.*

NAME: .....

ADDRESS:.....

CITY:.....

STATE:..... *POSTCODE*.....

PHONE.....

Mail to
MICRO 80
P.O. Box 213
Goodwood
South Australia 5034

☐ *I am interested, but I need more information.*

DEALER INQUIRIES INVITED